robotron

Systemunterlagendokumentation

MOS K1520

Anwenderdokumentation

Zusatzsoftware II

unter dem Betriebssystem SCPX 1526

Stand: 30.6.85

Anleitung fuer den Bediener fuer die "Zusatzsoftware II unter dem Betriebssystem SCPX 1526"

VEB Robotron Buchungsmaschinenwerk
Karl-Marx-Stadt 1985

Die vorliegende 1. Auflage der - Dokumentation "Zusatzsoftware II unter dem Betriebssystem SCPX 13526" - entspricht dem Stand Juni 1985 und unterliegt nicht dem Aenderungsdienst.

Nachdruck, jegliche Vervielfaeltigung oder Auszuege daraus sind unzulaessig.

Die Dokumentation wurde durch ein Kollektiv des VEB Robotron Buchungsmaschinenwerk Karl-Marx-Stadt ausgearbeitet.
Im Interesse einer staendigen Weiterentwicklung werden die Leser gebeten, dem Herausgeber ihre Vorschlaege bzw. Hinweise zur Verbesserung mitzuteilen.

Anmerkung:

Die Anwenderdokumentation zum SCP-System besteht aus:

- Anleitung fuer den Bediener SCP 1520
- Anleitung fuer den Programmierer SCP 1520
  Teil I und Teil II (Sprachbeschreibung ASM)
- Anleitung fuer den Systemprogrammierer SCP 1520
- Hardwarebeschreibung
- Anwenderdokumentation BASIC-Interpreter
- Anwenderdokumentation BASIC-Compiler
- Anwenderdokumentation C-Compiler
- Anwenderdokumentation PASCAL
- Anwenderdokumentation FORTRAN
- Anwenderdokumentation Textverarbeitungssystem TP
- Anwenderdokumentation Installierungs-Programm fuer TP
- Anwenderdokumentation KOMBO-Druck
- Schulungshandbuch fuer das Textverarbeitungs-system TP
- Zusatzsoftware I
- Zusatzsoware II

VEB Robotron-Buchungsmaschinenwerk
Karl-Marx-Stadt
Software-Zentrum
9010 Karl-Marx-Stadt/PSF 129

INHALTSVERZEICHNIS Seite

1.	Treiber-Unterprogramme	7
1.1.	Treiber fuer den Anschluss mehrerer Handleseeinheiten HLE4 1520 (SCPX)	7
1.1.1.	Hardware	7
1.1.1.1.	Handleseinheit und Anschlusssteuerung	7
1.1.1.2.	Technische Daten der Handleseeinheit	7
1.1.1.3.	Fehlererkennung	8
1.1.1.4.	Die Plastkarte mit Magnetstreifen	8
1.1.1.4.1.	Aufbau der Karten	8
1.1.1.4.2.	Behandlung der Karten	9
1.1.2.	Anwenderschnittstelle	9
1.1.2.1.	Allgemeines	9
1.1.2.2.	Funktionsbeschreibung	9
1.1.3.	4-Bit-Code nach ISO 3554 fuer Spur 2	11
1.1.4.	Programmbeispiel fuer Verwendung des HLE4-Treibers	11
1.2.	Treiber fuer 1/2"-Magnetbandgeraet an A5130 MBG 1520 (SCPX)	15
1.2.1.	Hardware	15
1.2.1.1.	Anschlusssteuerung AMB K5025	15
1.2.1.2.	1/2"-Magnetbandgeraet CM 5300.01	15
1.2.1.3.	ISO 1863-1976	16
1.2.2.	Anwenderschnittstelle 1/2"-MB-Treiber	17
1.2.2.1.	Einstellen der Parameter	18
1.2.2.2.	Aufruf des Treibers	19
1.2.2.3.	Realisierung der Funktionen	20
1.2.2.4.	Rueckmeldung des Treibers	22
1.2.2.4.1.	Rueckmeldung im Statusbyte	23
1.2.2.4.2.	Rueckmeldung im Ergebnisvektor	26
1.2.3.	Bedien- und Anzeigeelemente des MBG CM 5300.01	27
1.3.	Treiber Zusatzdrucker mit IFSS-Interface ZUDRU 1520 (SCPX)	29
2.	Sortieren und Mischen von Dateien SORT 1520 (SCPX)	30
2.1.	Uebersicht	30
2.2.	Programmaufruf/Kommandoeingabe	31
2.2.1.	Kommando-Format	32
2.2.2.	Reihenfolge der Kommandoeingaben	32
2.2.3.	Aendern/Fortsetzen von Kommandos	33
2.2.4.	Kommando-Zeilen	33
2.2.5.	Kommando-Notation	33
2.3.	Kommandobeschreibung	34
2.3.1.	Eingabedatei(en)	34
2.3.1.1.	INPUT-ATTRIBUTES	34
2.3.1.2.	SORT-FILES	35

2.3.1.3.	MERGE-FILES	36
2.3.2.	Ausgabedatei	37
2.3.3.	Schluessel	39
2.3.4.	Start	42
2.4.	Kommandodatei	43
2.5.	Arbeitsdatei	43
2.6.	TAGSORT	44
2.7.	Ergebnisausschrift	44
2.8.	Weitere Kommandos	45
2.9.	Selektion	46
2.9.1.	Einfuehrung	46
2.9.2.	Bereichstest	47
2.9.3.	Weitere Selektionsmoeglichkeiten	48
2.9.4.	Selektion bei Nicht-ASCII-Daten	48
2.9.4.1.	Spezielle Testattribute	48
2.9.4.2.	Numerische Konstanten	49
2.9.4.3.	Konstantenliste	50
2.9.5.	Veraenderung der Rangfolge	51
2.10.	Anhang A: Formate	53
2.11.	Anhang B: Fehlerausschriften	61
3.	Universelles Dienstprogramm DIENST 1520 (SCPX)	68
3.1.	Einleitung	68
3.2.	Kommandoeingabesyntax des DIENST	71
3.2.1.	Das DIENST-Nummernmenue	71
3.2.2.	Kommandosyntax ohne Nummernmenue (mit Dateinamen)	71
3.2.3.	Kommandosyntax ohne Nummernmenue und ohne Dateinamen	72
3.3.	Steuertasten	72
3.3.1.	Steuertasten fuer Bildschirm	72
3.3.2.	Steuertasten fuer Kommandoeingabe	73
3.4.	Kommandos des DIENST	73
3.4.1.	? (Hilfsfunktion)	73
3.4.2.	CHECK	74
3.4.3.	CM	74
3.4.4.	COPY	75
3.4.5.	DIR	76
3.4.6.	DISK	77
3.4.7.	DS	77
3.4.8.	DUMP/DUMPX/DUMPH/DUMPA	79
3.4.9.	ERA	80
3.4.10.	EXIT	81
3.4.11.	FILL	81
3.4.12.	GO	82
3.4.13.	GROUP	82
3.4.14.	JP/EX	83
3.4.15.1.	LOAD	83
3.4.15.2.	SAVE	84
3.4.16.	LOG	85
3.4.17.	MOVE	86
3.4.18.	READ	87
3.4.19.	WRITE	88

3.4.20.	READGR	89
3.4.21.	WRITEGR	90
3.4.22.	RECLAIM	90
3.4.23.	REN	91
3.4.24.	RESET	92
3.4.25.	RUN	93
3.4.26.	SEARCH	94
3.4.27.1.	SET [-character]	95
3.4.27.2.	SET	96
3.4.27.3.	SET [+character]	96
3.4.28.	SETDIR/SETSYS	96
3.4.29.	SETRO/SETWR	97
3.4.30.	SORT	97
3.4.31.	SIZE	98
3.4.32.	SPEED	98
3.4.33.	STAT	99
3.4.34.	TEST	99
3.4.35.	TYPE/TYPEX/TYPEH/TYPEA	100
3.4.36.	USER	101
3.4.37.	XUSER	101
3.4.38.	UR	102
3.5.	Fehlermeldungen	102
4.	Komprimieren einer SCP-Datei CONF 1520 (SCPX)	105
4.1.	Einleitung	105
4.2.	Aufrufen des Programmes	105
4.3.	Eingabe der Parameter der Quelldatei	105
4.4.	Eingabe der Parameter der Zieldatei	106
4.5.	Programmende	109
4.6.	Fehlerausschriften	109
4.7.	Zusammenhaenge des Dateiaufbaus SIOS-Datei und SCP-Datei bei der Bearbeitung mit UTR1 und CONF	110
5.	Drucken einer SCP-Datei PRIF 1520 (SCPX)	112
5.1.	Einleitung	112
5.2.	Aufrufen des Programmes	112
5.3.	Eingeben der Dateiparameter	112
5.4.	Eingeben der Parameter fuer die Druckliste	113
5.5.	Dateiende	115
5.6.	Fehlermeldungen	115
6.	Wandeln einer SIOS-Textdatei fuer die TP-Bearbeitung TRTX 1520 (SCPX)	117
6.1.	Einleitung	117
6.2.	Aufrufen des Programmes	117
6.3.	Eingabe der Parameter der Quelldatei	118
6.4.	Eingabe der Parameter der Zieldatei	118
6.5.	Dateiende	120
6.6.	Fehlernachrichten	120

7.	Diskettenanalyseprogramm LESD 1520 (SCPX)	122
8.	Transformationsprogramm fuer Text 20 - Dateien in Text 30 - Dateien UTR3 1520 (SCPX)	125
8.1.	Einleitung	125
8.2.	Parametereingabe fuer die Text 20 - Diskette	125
8.3.	Parametereingabe fuer die Text 30 - Diskette	127
8.4.	Fehlernachrichten und Meldungen	129
8.4.1.	Fehlernachrichten	129
8.4.2.	Meldungen	131

1.___Treiber-Unterprogramme

1.1._Treiber_fuer_den_Anschluss_mehrerer_Handleseeinheiten HLE4_1520_(SCPX)

1.1.1._Hardware

1.1.1.1._Handleseeinheit_und_Anschlusssteuerung

Die Handleseeinheit (HLE) K 6503 ist ein Geraet zum Verarbeiten von Plastkarten mit Magnetstreifen und an den BC A5120/30 bzw. an Terminals anschliessbar.
Der Anschluss kann mit folgender Anschlusssteuerung erfolgen:

K 6003.10	fuer 1-4 HLE	Anschlusslaenge max.5m
K 6003.11	fuer 1-2 HLE	Anschlusslaenge max.5m
K 6003.20	fuer 1-2 HLE	Anschlusslaenge 25m

Die Anschlusssteuerung fuer den Handleser K 6003.10/.11/.20 dient der Kopplung der HLE K 6503.10 an die ZRE 2526 ueber das Linieninterface BUS K 1520 und realisiert dabei den Datenaustausch.
Es sind ueber je einen 10-poligen Steckverbinder 4 Handleser anschliessbar, die simultan bedient werden.

1.1.1.2._Technische_Daten_der_Handleseeinheit

Datentraeger:	Plastkarte mit Magnetstreifen nach TGL 42091,42092,42093,ISO 3554, DIN 9781,9785
Kopf:	1-Spur-Wiedergabekopf
Funktionsgruppen:	-Kartenfuehrung mit Kopfhalterung -Verstaerker -Takt-Datentrennung
Schreibkapazitaet:	Spur2: 37 numerische Zeichen
Schreibverfahren:	Wechseltaktschrift nach DIN 66010
Aufzeichnungscode:	4-Bit-Code (siehe Anlage 1)
Schreibdichte:	3,0 Bit/mm
Kartengeschwind.:	10...100 cm/s
Abmessungen:	56 mm Hoehe 75 mm Breite 180 mm Tiefe
Masse:	1,0 kg
Leistungsaufnahme:	max. 5 W

Speisespannung: + 5 V +/- 5%
+12 V +/- 5%

Die Geraete werden als Beistellbaugruppen (ohne eigene Stromversorgung) ueber ein Kabel (Zubehoer) an die jeweilige Anschlusssteuerung angeschlossen.

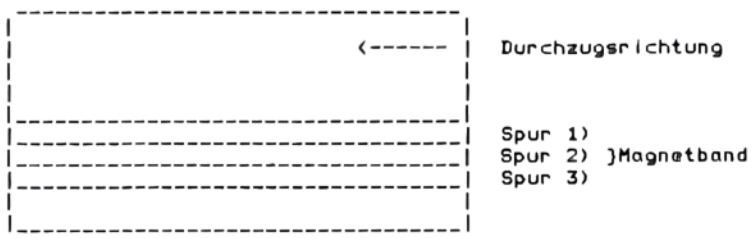
1.1.1.3. Fehlererkennung

Beim Lesen der Karten erfolgt die Kontrolle durch die Anschlusssteuerung K 6003. Jedes aufgezeichnete Datenzeichen wird durch ein Paritaetsbit auf ungerade Paritaet ergaenzt.
Jeder Datensatz wird durch ein LRC-Zeichen abgeschlossen, das alle Bits gleicher Wertigkeit, ausser dem Paritaetsbit, auf gerade Paritaet ergaenzt.
Das LRC-Zeichen selbst wird auf ungerade Paritaet ergaenzt.

1.1.1.4. Die Plastkarte mit Magnetstreifen

1.1.1.4.1. Aufbau der Karten

Als Datentraeger wird die Plastkarte mit Magnetstreifen nach ISO 3554 verwendet.
Die Karte besteht aus Kunststoff oder kunststofflaminiertem Werkstoff und ist einseitig mit einem Magnetstreifen versehen.
Mit der HLE kann nur die Spur 2 gelesen werden.



Die Spur 2 wird im 4-Bit-Code nach ISO 3554 aufgezeichnet (d.h. ein Zeichen entspricht einem 4-Bit-Element). Die Spur ist folgendermassen aufgebaut:

1	Startzeichen
37	numerische Zeichen
1	Stopzeichen
1	Pruefzeichen

1.1.1.4.2. Behandlung der Karten

Sichtbare Verschmutzungen der Kartenoberflaeche sind vor der Eingabe in die HLE zu entfernen.
Verletzungen der Magnetstreifenoberflaeche sind auf alle Faelle zu vermeiden.
Zum Schutz vor Verschmutzung und mechanischer Beschaedigung koennen die Magnetsreifen mit einer mehrere Mikrometer dicken Deckschicht aus durchsichtigem, nicht magnetisierbarem Material versehen sein.
Vor fremden Magnetfeldern sind die Karten zu schuetzen.

1.1.2. Anwenderschnittstelle

1.1.2.1. Allgemeines

Der Treiber wird als REL-Datei bereitgestellt (verschiebliche Objektcodedatei).
Der Treibername ist HLE. Dieses Symbol ist vom Anwender als externes Symbol zu definieren.

Der Treiber ist mit **Call HLE** aufzurufen.

Der Treiber rettet keine Register. Dies ist bei Bedarf vor Aufruf des Treibers vom aufrufenden Modul zu gewaehrleisten.
Der Treiber arbeitet im Interruptbetrieb (INT-Mode 2) und belegt innerhalb der Interruptsaeule (0F740H - 0F7DFH) 16 Byte.

1.1.2.2. Funktionsbeschreibung

Der Aufruf der Funktionen des Treibers erfolgt nach folgendem Schema:

```
Register  C - Funktion 0...n
              (Betriebsbeginn mit Funktion 0)
Register DE - Paramater ( HLE NR. )
Register HL - Parameter ( Einleseadresse fuer Fkt.1 )
CALL          HLE
```

Es sind folgende 4 Funktionen aufrufbar :

```
Funktion 0  - Initialisieren der HLE
Funktion 1  - Lesen
Funktion 2  - Statusabfrage
Funktion 3  - Loeschen der LED-Anzeige
```

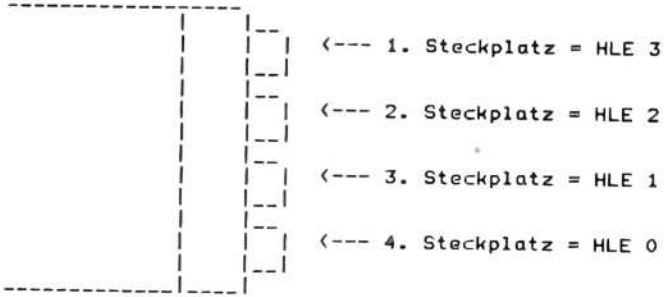
Fuer jede Funktion werden nachfolgende Registerzuweisungen vereinbart:

Register	Wirkung der LED-Anzeige	NR. der Funktion / Wirkung
C=0 E=0...3	Aus	0 HLE in Grundstellung (Initialisierung) HLE NR.
C=1 E=0...3 HL=...	Ein	1 Leseanforderung an HLE HLE NR. Einleseadresse
C=2 E=0...3	Ein	2 Statusabfrage HLE NR.
C=3 E=0...3	Ein	3 Loeschen der LED (als Quittierung des Lesevorganges durch das Anwenderprogramm) HLE NR.

Statusmeldungen bei Statusabfrage HLE :

Register	Bedeutung
A = FF	HLE nicht initialisiert bzw. keine Leseanforderung initialisiert
A = FE	Lesen initialisiert ,aber noch nichts gelesen
A = FD	Lesen initialisiert ,angefangen mit Lesen
A = 7F	fertig mit Lesen / Fehler
A = 00	fertig mit Lesen / kein Fehler

Steckplaetze fuer 4 Handleseeinheiten (Seitenansicht K 6003):



Die Funktionen 0,1,2 koennen in beliebiger Reihenfolge aufgerufen werden. Ist die HLE bei Aufruf C=1 noch nicht initialisiert, so wird dies vom Treiber automatisch ausgefuehrt.
Mit der Funktion 3 hat der aufrufende Modul die Moeglichkeit, die Lesequittierung selbst zu steuern.
Wird die Funktion 3 nicht aufgerufen, so bleibt die LED nach Initialisierung staendig aktiv (LED leuchtet).
Mit Funktion 3 kann der Anwender nach Pruefungen (o.ae.) die LED loeschen.
In der Initialisierungsfunktion wird die SCPX-Version 1.1. getestet. Ist der Test negativ, so erscheint am Bildschirm die Meldung: "Sorry SCPX 1.1.(or upward) is required !"
Gleichzeitig erfolgt ein Programmabbruch (Warmstart).

1.1.3. 4-Bit-Code nach ISO 3554 fuer Spur 2

Bits	Codewert	Zeichen
a_3 a_2 a_1 a_0		
0 0 0 0	0	0
0 0 0 1	1	1
0 0 1 0	2	2
0 0 1 1	3	3
0 1 0 0	4	4
0 1 0 1	5	5
0 1 1 0	6	6
0 1 1 1	7	7
1 0 0 0	8	8
1 0 0 1	9	9
1 0 1 0	10	wird nicht genutzt
1 0 1 1	11	Startzeichen
1 1 0 0	12	wird nicht genutzt
1 1 0 1	13	Leerfeldzeichen
1 1 1 0	14	wird nicht genutzt
1 1 1 1	15	Endezeichen

1.1.4. Programmbeispiel fuer Verwendung des HLE4-Treibers

```
;
;Programmbeispiel:
;
;Das nachfolgende Programm zeigt den von einer vorher aus-
;gewaehlten Handleseeinheit (0...3) gelesenen Inhalt einer
;Karte auf.
;Beim Lesen einer Karte wird nacheinander angezeigt:
;                    STATUS  HLE(NR.):
;                    DATEN   HLE(NR.):
```

```
aseg
org     100h
bdos    equ 0005h
wstart  equ     0h                      ;Warmstartadresse
extrn   hle

wied:   ld      de,tex0                 ;Ausgabe Kopf
        ld      c,9
        call    bdos
wh:     ld      de,tex1                 ;Anweisung Taste druecken
        ld      c,9
        call    bdos
        ld      c,10
        ld      de,mkp                  ;HLE-NR. auf Merkplatz mkp
        call    bdos
        ld      a,(mkp+2)

; Pruefung auf gueltige Eingabe fuer HLE NR.(0...3)

        ld      b,'4'
        cp      b
        jr      nc,wied
        ld      b,'0'
        cp      b
        jr      c,wied                  ;keine gueltige Eingabe

; Funktion Initialisieren einstellen

        xor     30h
        ld      e,a
        ld      (mke),a
        ld      c,0
        call    hle                     ;Aufruf HLE Treiber

; Funktion Lesen einstellen

        ld      hl,mke
        ld      e,m                     ;E :=HLE NR.
        ld      hl,buffx                ;HL:=Lesepufferadresse
        ld      c,1
        call    hle                     ;Aufruf HLE Treiber

; Abbruch des Testprogrammes ?
rs:     ld      c,6                     ;Direkt Console Input
        ld      e,0ffh
        call    bdos
        cp      3                       ;Ist Control-C Taste gedrueckt?
        jp      z,wstart                ;Ja-->Warmstart / Abbruch

; Funktion Statusabfrage einstellen

        ld      hl,mke
        ld      e,m                     ;E :=HLE NR.
        ld      c,2
        call    hle                     ;Aufruf HLE Treiber
```

```
; Lesen beendet ?

        bit     7,a             ;Bit7=0?
        jr      z,aus
        jp      rs              ;Lesen nicht beendet/Warten

; Ausgabe der gewaehlten HLE NR.

aus:    ld      hl,mke
        ld      e,m
        push    de
        push    af
        ld      a,'0'
        add     a,e
        ld      (shle),a        ;NR der HLE f.Status
        ld      (dhle),a        ;NR der HLE f.Daten

; Ausgabe der Statusmeldung

        ld      de,tex2         ;'STATUS:'
        ld      c,9
        call    bdos
        pop     af
        ld      de,teok         ;'00'
        or      a
        jr      z,out
        ld      de,tef          ;'7F***FEHLER***'
out:    ld      c,9
        call    bdos

; Ausgabe 'Daten:'

        ld      de,tex3         ;'DATEN:'
        ld      c,9
        call    bdos

; Funktion LED loeschen einstellen

        pop     de
        ld      c,3
        call    hle             ;Aufruf HLE Treiber
; Ausgabe der Daten aus dem Lesepuffer

        ld      de,buffx
        ld      c,9
        call    bdos
        jp      wh
```

```
tex0:   db      12,10,13,'**********************************'
        db      10,13,   '*                *                 *'
        db      10,13,   '*                                  *'
        db      10,13,   '*   * TESTPROGRAMM fuer HLE 4 *    *'
        db      10,13,   '*                                  *'
        db      10,13,   '*                *                 *'
        db      10,13,   '**********************************'
        db      10,13,24h
tex1:   db      10,10,10,13,'==> GEBEN SIE DIE NR.'
        db      'DER HLE EIN! (0...3)',24h
mkp:    db      1,0,0
mke:    db      0
buffx:  ds      37,0h
        db      24h
tex2:   db      10,10,10,13,'STATUS HLE'
shle:   db      '1: ',24h
tex3:   db      10,13,'DATEN  HLE'
dhle:   db      '1: ',24h
tef:    db      '7F***FEHLER****'
teak:   db      '00*'
end
```

1.2. Treiber fuer 1/2"-Magnetbandgeraet an A5130 MBG 1520 (SCPX)

1.2.1. Hardware

1.2.1.1. Anschlussteuerung AMB K 5025

Die Anschluss-Steuerung AMB K 5025 dient zum Anschluss des 1/2"-Magnetbandgeraetes CM 5300.01 an den Buerocomputer A5130 ~~(BC)~~.

Bestandteile:

Anschlussplatte:	045-8071/81	DMA-Steckeinheit
Steuereinheit:	012-7102	ZRB-K 2522
	045-8070	Dekodiersteckeinheit
	045-8069	CRC-Steckeinheit
	045-8068	Koppelsteckeinheit

1.2.1.2. 1/2"-Magnetbandgeraet CM 5300.01

Das Magnetbandgeraet CM 5300.01 ist ein Massenspeicher fuer sequentiellen Zugriff. Es kann auf Grund seiner Eigenschaften als externer Speicher fuer grosse Datenmengen dienen oder als Zwischendatentraeger fuer Grossrechner eingesetzt werden.
Durch den Anschluss des Magnetbandgeraetes an den BC ist die Konvertierung von Minidisketten 5,25", Standarddisketten 8" einfacher oder doppelter Bitdichte und Magnetbandkassetten auf 1/2"-Magnetband gewaehrleistet.

Technische Daten:

Aufzeichnungverfahren:	9 Spuren (ISO 1863-1976)
Aufzeichnungsdichte:	32 Bit/mm (800 bpi)
Uebertragungsgeschwind.:	10 kByte/s
Bandgeschwindigkeit:	0,3175 m/s
Rueckspulgeschwindigkeit:	1,5 m/s
Magnetband: -Laenge:	370 m
-Breite:	12,7 mm

Grundfunktionen:

Blockschreiben

Blocklesen

Schreiben von 2 Bandmarken u. Ruecksetzen in die Luecke dazwischen

Vor-/Ruecksetzen um Bloecke

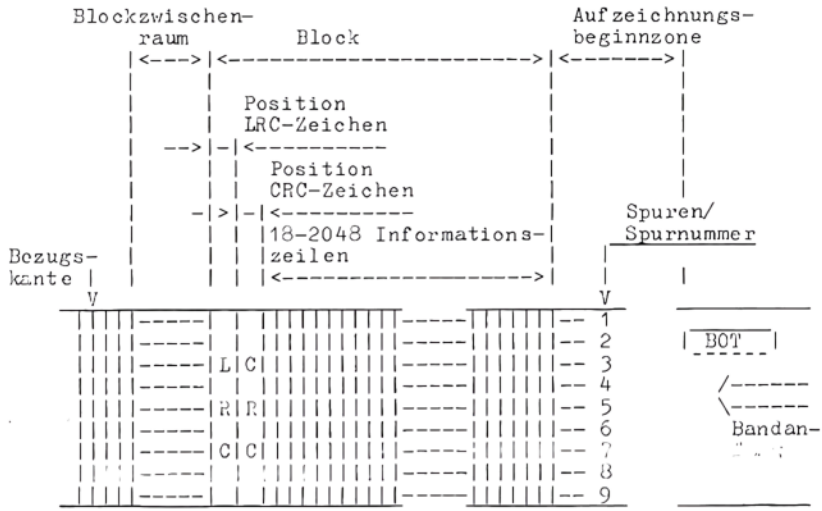
Vor-/Ruecksetzen um Bandmarken (Dateien)

Rueckspulen

1.2.1.3. ISO 1863-1976

Dieser Internationale Standard trifft Festlegungen ueber das Format und die Aufzeichnung eines 1/2"-Magnetbandes.

Magnetband - Spurendarstellung



-----> Vorlaufrichtung des Bandes
<----- Rueckla ufrichtung des Bandes

"|||...||" - Datenbits

- Identifizierung der Spuren:

Spur des Magnetbandes:	1	2	3	4	5	6	7	8	9
Wertigkeit:	2^3	2^0	2^4	p	2^5	2^6	2^7	2^1	2^2

p ist das Pruefbit. Die Zahlenparitaet ist ungeradzahlig. Die anderen 8 Bit bilden eine Informationszeile.

- Festlegungen:

Breite einer aufgezeichneten Spur:	>	1.09 mm	
Zeilenabstand:		0,03175 mm	+3%
			-3%
Spurenabstand:		1,5 mm	
Blockzwischenraum:	>	12,7 mm	
	<	7,6 m	
Aufzeichnungsbeginnzone:	>	76,0 mm	
	<	7,6 m	

Aufzeichnungsdichte:	32	Zeilen/mm
Aufzeichnungsgeschwindigkeit:	31,75	cm/s

Reflektormarken

Jede Bandspule besitzt 2 lichtreflektierende Marken.

Bandanfangsmarke	**BOT** :	- Abstand 4,9 +0,6 -0,6 m vom Bandanfang
Bandendemarke	**EOT** :	- Abstand 7,6 +1,5 0 m vom Bandende

Bandmarke

Zur Trennung der Dateien wird ein einzelner Kontrollblock (BM) geschrieben. Dieser Block wird von einem LRC-Zeichen begleitet. Die Bandmarke wird durch "1" in den Spuren 2, 3 und 8 dargestellt (alle anderen Spuren 0).

Blockaufbau

Jeder Block besteht aus 18 bis 2048 (2 K Byte) Informationszeilen. Am Ende jedes Blockes wird das CRC-Zeichen (Kontrollzeile der zyklischen Redundanz) zur Fehlererkennung auf einer Spur geschrieben.
Im Anschluss an das CRC-Zeichen wird das LRC-Zeichen (Laengspruefzeile) zur spaeteren Fehlererkennung beim Lesen aufgezeichnet. Ein Laengspruefzeichen wird auf den Spuren markiert, deren Zeichenanzahl ungerade ist.
Mit dem LRC-Zeichen ist jedoch nur eine geringe Fehlerrate erkennbar.

1.2.2. Anwenderschnittstelle 1/2"-MB-Treiber

Festlegungen:

- Der Treiber wird als verschiebliche Datei (Typ .REL) bereitgestellt. Er ist im absoluten Modus mit ORG-Anweisung programmiert worden.

```
ASEG
ORG 103H
```

- Es ist keine Interruptarbeit erforderlich. Die AHB arbeitet im Batch-Betrieb.

- Dem Treiber muessen vom Anwender folgende Parameter in den CPU-Registern bereitgestellt werden:

  in Register C - **Funktionscode FC**
  in Register DE - **Blockadresse BA**
  in Register HL - **Blocklaenge BL**

- Rueckmeldungen des Treibers an das aufrufende Programm werden

in Register A - **Statusbyte**
in Register HL - **Ergebnisvektor** bereitgestellt.

- Der Name des Treibers MBG ist fuer das aufrufende Programm das externe Symbol.

- Das aufrufende Programm muss mit **END <Progr.-anf.-marke>** beendet werden, damit der Programmverbinder auf die Adr. 100H einen Sprung schreibt, der auf den logischen Programmanfang zeigt (auch mit G-Schalter moeglich - siehe LINK).

1.2.2.1. Einstellen der Parameter

Vor Aufruf der einzelnen Funktionen des Treibers sind folgende Parameter einzustellen:

In Register C - Funktionscode FC

- FC fuer die einzelnen Funktionen:

Funktion	FC
Schreiben	01H
Lesen	02H
BK schreiben	04H
Block vorsetzen	08H
Block ruecksetzen	88H
Datei (BK) vorsetzen	10H
Datei (BK) ruecksetzen	90H
Rueckspulen	20H

- Bitbelegung des Funktionscodebytes:

0-vor setzen 1-rueck-setzen	0-1-ohne Bedeutung	0-1-rueck-spulen	0-1-BM-Steuerbit	0-1-Block Steuerbit	0-1-BM schr.	0-1-Block lesen	0-1-Block schr.
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

in Register Blockadresse BA

Die Blockadresse BA muss bei folgenden Funktionen vom Anwender bereitgestellt werden:

- Lesen:

BA ist die Adresse im Speicher, auf die die vom Magnetband gelesenen Daten uebernommen werden (E/A-Bereich).

- Schreiben:

BA ist die Adresse im Speicher, von der die Daten auf das Magnetband geschrieben werden (E/A-Bereich).

in Register HL - Blocklaenge BL

Die Blocklaenge BL wird bei folgenden Funktionen benoetigt:

- Lesen:

BL ist die Anzahl der zu lesenden Bytes.

- Schreiben:

BL ist die Anzahl der zu schreibenden Bytes.

- Block vor/rueck:

BL ist die Anzahl der Bloecke, die vor- oder rueckgesetzt werden sollen.

- Datei vor/rueck:

BL ist die Anzahl der Dateien, die vor- oder rueckgesetzt werden sollen.

1.2.2.2. Aufruf des Treibers

Der Aufruf des Treibers erfolgt nach Parametereingabe durch **CALL MBG**.

```
LD   C ,FC
LD   DE,BA
LD   HL,BL
CALL MBG
```

1.2.2.3. Realisierung der Funktionen

- Block schreiben

Der durch die Blockadresse BA gekennzeichnete Speicherbereich wird ohne Zwischenspeicherung mit Datensicherungsmassnahmen auf Band geschrieben.
Die Aufzeichnungskontrolle erfolgt durch Bildung des CRC-Zeichens fuer Lesedaten. Werden Aufzeichnungsfehler erkannt, werden 3 weitere Aufzeichnungsversuche auf der gleichen Bandstelle und je 4 auf 3 um ca. 100 mm vorgezogenen und geloeschten Bandstellen unternommen. Bei negativem Ergebnis wird Bit7 im Statusbyte gesetzt.
Das Aufzeichnungssystem befindet sich nach Abbruch vor dem fehlerhaft aufgezeichneten Block, so dass dieser anschliessend ueberschrieben werden kann.
Befindet sich die BOT-Marke unter der Abtasteinrichtung, erfolgt durch die ANB automatisch die Aufzeichnung der Initialluecke (400 mm) vor dem Aufzeichnen des Datenblocks.
Bei Erkennen der BOT-Marke wird Bit4 im Statusbyte gesetzt.
Ist keine Magnetbandspule eingelegt oder der Ring zum Aufzeichnungsschutz abgenommen, wird im Statusbyte Bit6 gesetzt.

- Block lesen

Die Bytes werden vom Band gelesen und in den durch die Blockadresse BA gekennzeichneten Speicherbereich eingetragen. Die gelesene Bytezahl wird im Ergebnisvektor angegeben.
Ist der Block auf dem Magnetband laenger oder kuerzer als der zugewiesene Speicherbereich, werden die restlichen Bytes gezaehlt und Bit2 im Statusbyte markiert. Im Ergebnisvektor wird die tatsaechlich gelesene Byteanzahl von der ANB eingetragen.
Die Daten sind erst nach gegebener Quittung gueltig.
Die Datensicherung bei der Wiedergabe erfolgt durch Bildung und Kontrolle des CRC- und des LRC-Zeichens.
Bei Wiedergabefehler werden bis zu 8 Wiedergabeversuche unternommen. Bei negativem Ergebnis wird Bit7 im Statusbyte gesetzt. Das Wiedergabesystem befindet sich dabei vor dem fehlerhaften Block, d.h. bei erneutem Lesekommando wird der Lesevorgang wiederholt. Der fehlerhaft gelesene Block steht im Speicherbereich zur Verfuegung. Er kann durch das Kommando "Block vorsetzen" ueberlesen werden.

Wird beim Wiedergeben eine Bandmarke erkannt, dann wird vor die Bandmarke positioniert und im Statusbyte Bit5 belegt. Die Bandmarke kann durch das Kommando "Block vorsetzen" oder "Datei vorsetzen" ueberlesen werden.
Die BOT-Marke wird analog dem Aufzeichnungsvorgang behandelt.

- Bandmarke schreiben

Es werden 2 Bandmarken nach ISO 1863 aufgezeichnet. Danach wird das Band zwischen beide Bandmarken positioniert, um die Aufzeichnung einer weiteren Datei zu ermoeglichen.
Die Fehlerbehandlung erfolgt wie beim Block schreiben.
Da beim Ruecksetzen um 1 Bandmarke diese zur Kontrolle gelesen wird, ist im Statusbyte bei korrekter Ausfuehrung Bit5 gesetzt.
Ist keine Magnetbandspule eingelegt oder der Aufzeichnungsring abgenommen, wird Bit 6 im Statusbyte gesetzt.

- Vor-/Ruecksetzen um Bloecke

Das Kommandobyte "Block vor-/ruecksetzen" bewirkt die Funktion der Blockpositionierung mit Arbeitsgeschwindigkeit. Entsprechend der vorgegebenen Blockanzahl BL wird das Magnetband

- um n Bloecke vorgesetzt, wenn Bit7 von BL (HIGH) Null ist
  oder
- um n Bloecke rueckgesetzt, wenn Bit7 von BL (HIGH) 1 ist.

Der Lesevorgang wird zeitlich ueberwacht. Bei Ueberschreiten der zulaessigen Maximalluecke (>7500 mm) wird Bit7 im Statusbyte gesetzt.
Nach Funktionsausfuehrung wird der Ergebnisvektor versorgt und das Statusbyte aktualisiert. Entspricht die Anzahl der vor-/rueckgesetzten Bloecke nicht der durch BL geforderderten, dann wird Bit2 im Statusbyte gesetzt und die tatsaechlich vor-/rueckgesetzte Blockanzahl in den Ergebnisvektor eingetragen.

a) Bedingungen beim Block vorsetzen:

Wird eine Bandmarke erkannt, wird die Funktion abgebrochen. Das Magnetband steht vor dem 1. Datenblock der naechsten Datei in Vorwaertsrichtung. Die aktuelle Anzahl der vorgesetzten Bloecke wird im Ergebnisvektor angegeben. Im Statusbyte wird Bit5 belegt.
Bei Erkennen der EOT-Marke wird die Funktion fortgefuehrt und im Statusbyte Bit4 belegt.

b) Bedingungen beim Blockruecksetzen:

Wird eine Bandmarke erkannt, wird die Funktion abgebrochen. Das Magnetband steht vor der Bandmarke in Vorwaertsrichtung. Bei Erkennen der BOT-Marke wird die Funktion ebenfalls abgebrochen.
In beiden Faellen wird die aktuelle zurueckgesetzte Blockanzahl in den Ergebnisvektor eingetragen und im Statusbyte Bit3 bzw. Bit5 gesetzt.

Das Magnetband wird um n Bandmarken vor- bzw. rueckgesetzt. Wird beim Vorsetzen das logische Bandende erkannt (zwei durch Blockluecke getrennte Bandmarken), wird die Funktion abgebrochen, die Anzahl der vorgesetzten Dateien im Ergebnisvektor bereitgestellt und im Statusbyte Bit5 belegt. Danach wird das Magnetband zwischen beide Bandmarken positioniert.
Wird beim Ruecksetzen die BOT-Marke erkannt, wird die Funktion abgebrochen und im Ergebnisvektor die Anzahl der rueckgesetzten Dateien bereitgestellt. Im Statusbyte wird Bit3 gesetzt. Nach Dateiruecksetzen steht das Magnetband in Vorwaertsrichtung vor dem 1. Block der Datei.
Wird bei Operationen die BOT-Marke erreicht, wird im Statusbyte zusaetzlich Bit4 gesetzt.

Bei Nichtuebereinstimmung von geforderter Dateianzahl und tatsaechlicher Anzahl wird Bit2 im Statusbyte gesetzt und die aktuelle vor-/rueckgesetzte Dateianzahl in den Ergebnisvektor eingetragen.
Da die Funktion im Normalfall mit Erkennen einer Bandmarke abgeschlossen wird, ist im Statusbyte Bit5 belegt.

- Rueckspulen

Das Magnetband wird bis zur BOT-Marke zurueckgespult. Das Ergebnis wird im Statusbyte bereitgestellt.
Der Umspulvorgang wird durch die AMB zeitlich ueberwacht. Bei Ueberschreiten der Umspulzeit wird Bit7 im Statusbyte gesetzt. Das Umspulen wird nach Erreichen des Bandanfangs quittiert.

1.2.2.4. Rueckmeldungen des Treibers

Nach Ausfuehrung der Funktion gibt der Treiber folgende Rueckmeldungen:

- in Register A - **Statusbyte** (siehe Pkt. 2.2.)
- in Register HL - **Ergebnisvektor** (Anzahl der gelesenen Bytes bzw. Anzahl der vor-/rueckgesetzten Bloecke/Dateien)

1.2.2.4.1. Rueckmeldung im Statusbyte

Bitbelegung des Statusbytes:

0-1-R/W-Fehler Geraetefehler	0-1-Aufzeichnungsverletz.	0-1-BM erkannt	0-1-EOT erkannt	0-1-BOT erkannt	0-BL=Ergebnisvektor 1-ungleich	0-MBG in Rohe 1-MBG arbeitet	0-MBG bereit 1-MBG nicht bereit
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Die ersten beiden Bits sind nach Punktionsausfuehrung immer mit Null belegt.

Bit 0 = 0
Bit 1 = 0

Der Treiber wartet intern immer auf Funktionsbeendigung, d.h. Bit1 = 0 ist nach Rueckkehr aus dem Treiber immer eingestellt. Sollte das MB-Geraet nicht bereit sein (nicht eingeschalten, kein Band eingelegt), so ist Bit0 = 1 gesetzt.

Statusmeldungen der einzelnen Funktionen

- Schreiben:

Code im Statusbyte	Bedeutung
00H	- Funktion ausgefuehrt
10H	- EOT-Marke erkannt
Fehlermeldungen:	
40H	- Aufzeichnungsverletzung (Aufzeichnungsschutzring abgenommen oder keine MB-Spule eingelegt)
80H	- falscher Funktionscode - Aufzeichnungsfehler - Geraetefehler

- Lesen:

Code im Statusbyte	Bedeutung
00H	- Funktion ausgefuehrt

Fehlermeldungen:	
04H	- vorgegebene Byteanzahl BL ungleich tatsaechlich gelesene Byteanzahl
14H	- EOT-Marke erkannt, vorgegebene Byteanzahl BL ungleich tatsaechlich gelesene Byteanzahl
24H	- BM erkannt, vorgegebene Byteanzahl BL ungleich tatsaechlich gelesene Byteanzahl
80H	- falscher Funktionscode - Wiedergabefehler - Geraetefehler - Ueberschreiten zulaessige Maximalluecke (7500 mm)

= Bandmarke (BM) schreiben:

Code im Statusbyte	Bedeutung
20H	- Funktion ausgefuehrt, BM erkannt

Fehlermeldungen:	
10H	- EOT-Marke erkannt
40H	- Aufzeichnungsverletzung (Aufzeichnungsring abgenommen oder keine MB-Spule eingelegt)
80H	- falscher Funktionscode - Aufzeichnungsfehler - Geraetefehler

= Block versetzen:

Code im Statusbyte	Bedeutung
00H	- Funktion ausgefuehrt

Fehlermeldungen:	
14H	- EOT-Marke erkannt, vorgegebene Blockanzahl BL ungleich tatsaechlich vorgesetzte Blockanzahl

24H	- BM erkannt, vorgegebene Blockanzahl BL ungleich tatsaechlich vorgesetzte Blockanzahl (Band steht nach BM, am Anfang der neuen Datei)
80H	- falscher Funktionscode - Geraetefehler - Ueberschreiten zulaessige Maximalluecke (7500 mm)

- Block ruecksetzen:

Code im Statusbyte	Bedeutung
00H	- Funktion ausgefuehrt
20H	- Funktion ausgefuehrt, BM erkannt

Fehlermeldungen:

0CH	- BOT-Marke erkannt, vorgegebene Blockanzahl BL ungleich tatsaechlich rueckgesetzte Blockanzahl
24H	- BM erkannt, vorgegebene Blockanzahl BL ungleich tatsaechlich rueckgesetzte (Band steht an Dateianfang)
80H	- falscher Funktionscode - Geraetefehler

- Datei vorsetzen:

Code im Statusbyte	Bedeutung
20H	- Funktion ausgefuehrt, BM erkannt

Fehlermeldungen:

14H	- EOT-Marke erkannt, vorgegebene Dateianzahl BL ungleich tatsaechlich vorgesetzte Dateianzahl

24H	- BM erkannt, vorgegebene Dateianzahl BL ungleich tatsaechlich vorgesetzte Dateianzahl (logisches Bandende)
80H	- falscher Funktionscode - Geraetefehler

- Datei ruecksetzen:

Code im Statusbyte	Bedeutung
20H	- Funktion ausgefuehrt, BM erkannt

Fehlermeldungen:

0CH	- BOT-Marke erkannt, vorgegebene Dateianzahl BL ungleich tatsaechlich rueckgesetzte Dateianzahl
80H	- falscher Funktionscode - Geraetefehler

- Rueckspulen:

Code im Statusbyte	Bedeutung
08H	- Funktion ausgefuehrt, BOT-Marke erkannt

Fehlermeldungen:

80H	- falscher Funktionscode - Geraetefehler

1.2.2.4.2. Rueckmeldungen im Ergebnisvektor

- Schreiben

Es erfolgt keine Rueckmeldung im Ergebnisvektor.

- Lesen

Von der AMB wird die Anzahl der gelesenen Bytes in den Ergebnisvektor eingetragen.

- BM schreiben

Es erfolgt keine Rueckmeldung im Ergebnisvektor.

- Block vor-/ruecksetzen

Von der AMB wird die Anzahl der vor-/rueckgesetzten Bloecke eingetragen.

- Datei vor-/ruecksetzen

Von der AMB wird die Anzahl der vor-/rueckgesetzten Dateien eingetragen.

- Rueckspulen

Es erfolgt keine Rueckmeldung im Ergebnisvektor.

1.2.3. Bedien- u. Anzeigeelemente des MBG CM 5300.01

- Netztaste:
  - bei Einschalten wird MB-Einheit angeschlossen

- Anzeige "Netz":
  - leuchtet bei eingeschalteter Stromversorgung

- Taste "Laden":
  - bringt den Pufferhebel in den Arbeitsbereich und stellt "on-line"-Zustand her
  - Band wird nach vorn gespult bis zum Erkennen der BOT-Marke
  - bei Erkennen dieser Marke leuchtet diese Taste auf
  - ist bei Netzeinschaltung Magnetband eingelegt, so ist nach "Laden" der Vorwaertslauf des MB durch "STOP" abzubrechen und durch "Rueckspulen" auf die Marke "BOT" zu positionieren

- Taste "on-line":

  ДИСТАНЦ

  - durch Druecken dieser Taste wird logische Verbindung zwischen Rechner und Magnetbandeinheit hergestellt
  - die Steuerung erfolgt ueber Interfaceleitungen, die Taste leuchtet auf
  - Umschaltung von "on-line" auf "off-line" durch Taste "STOP"

- Anzeige "Aufzeichnungsschutz":

  ЗАЩИТА ЗАПИСИ

  - leuchtet auf, wenn keine MB-Spule eingelegt ist oder wenn der Ring zum Aufzeichnungsschutz abgenommen ist

- Taste "Rueckspulen":

  ПЕРЕМОТКА

  - es erfolgt schneller Ruecklauf bis zur BOT-Marke
  - danach wird Operation "Entladen" ausgefuehrt
  - Pufferhebel gehen in Endlage zurueck und Band wird ganz abgespult
  - alle Operationen, ausser "STOP", werden blockiert

- Taste "STOP":

  СБРОС

  - Bandtransport wird unterbrochen, Interfaceleitungen blockiert
  - Geraet wird in Betriebsart "off-line" gebracht

- Taste "vorwaerts":

  ВПЕРЕД

  - Bandvorlauf bis zum Erkennen der EOT-Marke
  - waehrend der Betriebsart "on-line" ist diese Taste wirkungslos

- Taste "rueckwaerts":

  НАЗАД

  - Bandruecklauf bis zum Erkennen der BOT-Marke
  - waehrend der Betriebsart "on-line" ist diese Taste wirkungslos

1.3. Treiber fuer Zusatzdrucker mit IFSS-Interface ZUDRU 1520 (SCPX)

Dieses Unterprogramm dient zum Aufruf eines Zusatzdruckers mit IFSS-Anschluss und muss in Anwenderprogramme eingebunden werden. Mit Hilfe dieses Programmes kann eine Zeichenkette ausgedruckt werden. Innerhalb dieser Zeichenkette sind alle druckbaren Zeichen und Steuerzeichen des Druckers zugelassen.

Folgende Register muessen im Anwenderprogramm voreingestellt werden:

HL --> Adresse der auszugebenden Zeichenkette

BC --> Anzahl der auszugebenden Zeichen (0 kein Zeichen)

Der Aufruf des Unterprogrammes erfolgt durch:

```
CALL ZUDRU.
```

Rueckmeldung im Register A:

```
<A> = 00 (HEX) --> kein Fehler
<A> = FF (HEX) --> Druckerfehler
```

Im Fehlerfall leuchtet LED auf. Durch Betaetigen einer beliebigen Taste ist die Weiterarbeit fast immer moeglich (z.B. falsches Steuerzeichen fuer Drucker!). Bei Papierendemeldung und technischen Stoerungen des Druckers muss der Fehler erst behoben werden. Sonst erfolgt die Meldung:

```
<A> = FF (HEX).
```

Das Unterprogramm wird unter der Bezeichnung ZUDRU.REL bereitgestellt.

2. Sortieren und Mischen von Dateien SORT 1520 (SCPX)

SORT 1520 (SCPX) ist ein Dienstprogramm zum Sortieren und Mischen von Dateien, die auf Disketten gespeichert sind. Das Programm hat eine Laenge von 20 K Byte.

2.1. Uebersicht

Sortieren/Mischen

- Sortieren von maximal 32 Eingabedateien in eine Ausgabedatei;
- Mischen maximal 32 vorsortierter Dateien;
- Sortieren und Mischen von Dateien kann im gleichen Lauf erfolgen;
- unabhaengig vom Sortier-/Mischvorgang koennen eine Satzselektion, eine Dateikonvertierung und weitere Operationen ausgefuehrt werden.

Datei- und Satzformate

- Verarbeitung von SCP-Disketten-Dateien;
- Dateien koennen ASCII-, BCD- und/oder Binaerdaten enthalten;
- maximale logische Satzlaenge betraegt 4096 Zeichen;
- Dateien koennen Saetze mit fester oder variabler logischer Satzlaenge enthalten;
- die als Schluessel (Sortiermerkmal) festgelegten Felder koennen eine feste Position im Satz besitzen oder eine variable, wenn der Satz aus durch Kommas getrennten Feldern besteht.

Sortierschluessel

- 1 bis 32 Felder koennen als Sortierschluessel angegeben werden; der Schluessel mit der hoechsten Prioritaet wird zuerst angegeben;
- Schluesseldaten koennen als ASCII-Text, ASCII-numerisch oder binaer interpretiert werden;
- Zusaetzliche Moeglichkeiten bei der Schluesselinterpretation sind das Umwandeln von Klein- in Grossbuchstaben, das Ignorieren des hoechstwertigen Bits eines Bytes und andere.

Satzselektion

- Saetze koennen nach vorgegebenen Kriterien aus Eingabedateien selektiert werden;
- die Selektion wird durch Vergleich eines Feldes mit einem konstanten Wert, einer Folge konstanter Werte oder einem anderen Feld im gleichen Satz spezifiziert;
- Vergleichsoperatoren sind KLEINER GLEICH, GLEICH usw., aber auch ENTHALTEN und NICHT ENTHALTEN. Die Vergleiche koennen wahlweise mit den logischen Operatoren UND, ODER und EXCLUSIV-ODER kombiniert werden;
- Zur Satzselektion koennen Felder fester oder variabler Position verwendet werden.

weitere Moeglichkeiten

- Teildateien:
  fuer jede Sortiereingabedatei koennen Anfangs- und End-Satznummer angegeben werden;

- Wechsel Ausgabediskette:
  ermoeglicht Sortieren in einem 2-Laufwerks-System ohne die Eingabedatei zu ueberschreiben;

- TAGSORT:
  Abhaengig von der zu sortierenden Datenmenge und der Schluessellaenge wird eine temporaere Disketten-Arbeitsdatei angelegt. Durch TAGSORT wird ein anderer Sortieralgorithmus verwendet, der zwar die Sortierzeit erhoeht, aber keine Arbeitsdatei erfordert bzw. den fuer die Arbeitsdatei benoetigten Diskettenbereich reduziert (2- bzw. 1-Laufwerkskonfigurationen);

- Konsolen-Ausgabe:
  Auswahl unter sechs Moeglichkeiten der Ausschrift von Informationen ueber den Sortier-/Mischlauf;

- Zusatzfunktionen:
  Konvertieren einer Datei eines bestimmten Satztyps in eine Datei anderen Typs;
  Veraenderung der Satzlaenge, Ausblenden bzw. Umpositionieren von Feldern.

Zusaetzliche Ausgabemoeglichkeiten

- Nur-Schluessel:
  Ausgabedatei enthaelt nur die Sortierbegriffe;
  kann zum Aufbau einer Indexdatei, zur Summenbildung oder zur Selektion oder zum Umpositionieren von Saetzen genutzt werden;

- Satznummern:
  Ausgabedatei enthaelt nur die Satznummern der Eingabedateien im ASCII- oder Binaerformat;
  die entsprechenden Schluessel koennen wahlweise mit gespeichert werden;

- Zeiger:
  Ausgabedatei enthaelt Sektornummern und relativen Satzbeginn (Offset), mit oder ohne Schluesselspeicherung.

2.2. Programmaufruf/Kommandoeingabe

Die von SORT 1520 (SCPX) ausfuehrbaren Funktionen werden durch Kommandoeingaben aktiviert. Die Kommandos koennen einzeln, zusammengefasst zu einer Kommandodatei oder in der SORT aufrufenden Systemzeile eingegeben werden.
Jede Kommandozeile wird durch <ENTER> beendet.

Im einfachen Fall wird SORT in der Systemzeile durch

SORT

geladen und zeigt durch Anzeige eines "*" seine Bereitschaft zur Kommandoeingabe an.

Einige oder alle Kommandos koennen in der Systemzeile nach dem Programmnamen "SORT" eingegeben werden, z.B.:

A>SORT INP=80,CR;SO=EIN.DAT;OUT=AUS.DAT;KEY=#2,4;GO

Waehrend des Sortier-/Mischlaufes ist keine Tastatureingabe moeglich.

2.2.1. Kommando-Format

Jedes SORT-Kommando besteht aus einem Schluesselwort, das den Kommandotyp festlegt, gefolgt von Parametern. Abhaengig vom Kommandotyp bestehen die Parameter aus Dateinamen, Zahlen und Schluesselwoertern.

Schluesselwoerter bestehen aus Buchstaben und Bindestrich "-" und besitzen eine Lang- und mehrere Kurzformen.

Buchstaben koennen als Klein- oder Grossbuchstaben eingegeben werden. Im folgenden werden die Schluesselworte vorrangig in ihrer selbstdokumentierbaren Langform dargestellt. Waehrend diese Form leicht lesbar ist, benoetigt die Kurzform weniger Eingabeaufwand. Die Kurzform wird gebildet, indem Buchstaben und/oder der Bindestrich am Ende des Schluesselwortes weggelassen werden.

Als minimale Anzahl Buchstaben eines Schluesselwortes muessen eingegeben werden: ein oder zwei Buchstaben bei den meisten Kommandos; drei, wenn das Wort mit "NO" beginnt.

Drei oder mehr Buchstaben werden stets akzeptiert.
Z.B. sind folgende Bezeichnungen aequivalent:

INPUT-ATTRIBUTE
INPUT
IN
I

Leerzeichen koennen anstelle von Kommas oder dem Gleichheitszeichen nach dem Kommandoschluesselwort verwendet werden. Leerzeichen koennen beliebig zwischen den Worten eingefuegt werden.

2.2.2. Reihenfolge der Kommandos

Die einzelnen Kommandos fuer einen Sortier-/Mischlauf koennen in beliebiger Reihenfolge eingegeben werden. Es gibt zwei Aus-

nahmen: Schluessel muessen in der gewuenschten Folge eingegeben werden, das Kommando GO zum Start muss zuletzt angegeben werden.

2.2.3. Aendern/Fortsetzen von Kommandos

Werden Kommandoeingaben wiederholt, gilt die letzte Eingabe. Dies ermoeglicht das spaetere Aendern bereits eingegebener Kommandos.

Kommandos mit vielen Eintragungen (z.B. KEY=) koennen erweitert werden, wenn man anstelle des Gleichheitszeichens "=" das Pluszeichen "+" angibt.

2.2.4. Kommando-Zeilen

SORT-Kommandos werden ueblicherweise einzeln und in einer Zeile eingegeben. Jedoch sind auch folgende Varianten erlaubt:

- mehrere Kommandos in einer Zeile:
  Kommandos muessen durch Semikolon getrennt werden;
- Fortsetzung eines Kommandos in einer Folgezeile:
  Eingabe eines Ampersand "&" vor <ENTER>;
- Kommentare :
  ein Vertikalstrich "|" definiert den Rest der Zeile als Kommentar.

2.2.5. Kommando-Notation

Zur Definition der Kommandos gilt folgende Notation:

INPUT	Grossbuchstaben fuer Kommandoschluesselwoerter
dateiname	Kleinbuchstaben fuer zu ersetzende Begriffe
( )	Eingabezwang fuer eine der uebereinanderstehenden Moeglichkeiten
[ ] { }	Eingabe kann weggelassen werden keine Eingabe oder Eingabe der angefuehrten Alternativen in beliebiger Anzahl und Reihenfolge, getrennt durch Komma oder Leerzeichen
...	beliebige Anzahl von Wiederholungen der vorangestellten Parameter

2.3. Kommandobeschreibung

Eine ausfuehrliche Erlaeuterung der Datei- und Datentypen enthaelt Anhang A.

2.3.1. Eingabedatei(en)

Drei Kommandos werden zur Definition der Eingabedateien verwendet:

INPUT-ATTRIBUTES	spezifiziert die Satzlaenge, den Satztyp und andere fuer alle Eingabedateien gemeinsamen Attribute; muss stets angegeben werden
SORT-FILES	Namen der zu sortierenden Dateien, wahlweise Angabe einer Anfangs- und Endsatznummer bei Sortierung einer Teildatei
MERGE-FILES	Namen der zu mischenden Dateien

Es muss entweder ein SORT-FILES- oder ein MERGE-FILES-Kommando angegeben werden. Beide koennen benutzt werden, wenn im gleichen Lauf sortiert und gemischt werden soll.

2.3.1.1. INPUT-ATTRIBUTES

```
INPUT-ATTRIBUTES = satzlaenge  {FIXED-LENGTH}
                               {CR-DELIMITED}
                               {VARIABLE    }
                               {RELATIVE    }
                               {NO-SINGLE-Z }
                               {NO-ZZZ      }
                               {FPZZZ       }
```

satzlaenge:
ist eine Zahl zwischen 1 und 4096, die die Laenge von Saetzen mit fester Laenge in Bytes angibt oder die maximale Laenge von Saetzen des Typs VARIABLE oder CR-DELIMITED.

Eines der folgenden vier Attribute kann zur Spezifizierung des Satztyps fuer alle Eingabedateien angegeben werden. Wird keines angegeben, wird als Standard FIXED-LENGTH angenommen.

FIXED-LENGTH	die Eingabedatei(en) bestehen aus Saetzen fester Laenge
CR-DELIMITED	die Eingabedatei(en) enthalten Saetze, normalerweise von variabler Laenge, die durch carriage return (0DH) und/oder line feed (0AH) getrennt sind
VARIABLE	die Eingabedatei(en) enthalten Saetze variabler Laenge mit der Angabe der Laenge in den ersten zwei Bytes jedes Satzes

Beachte:
Die Satzlaenge muss um 2 groesser angegeben werden als die maximale Anzahl der eigentlichen Datenzeichen des Satzes.

RELATIVE — die Eingabedatei(en) bestehen aus Saetzen des Typs RELATIVE

Beispiele:

```
INPUT-ATTRIBUTES = 128              (feste Laenge angenommen)
INPUT-ATTRIBUTES =  80,RELATIVE
INPUT-ATTRIBUTES =  32,CR-DELIMITED
INPUT-ATTRIBUTES = 256,FIXED-LENGTH
INPUT = 256,FIXED
I 256,FI                            (minimale Kurzform)
```

Die letzten drei Beispiele sind aequivalent.

Die folgenden drei INPUT-ATTRIBUTES definieren die Datei-Ende-Bedingung in FIXED-LENGTH-Dateien, die Binaerdaten enthalten. Fuer andere Datei- bzw. Datentypen sind diese Attribute ohne Bedeutung.

NO-SINGLE-Z — die PIXED-LENGTH-Datei ist durch **ein** 1AH (CTRL Z) beendet

NO-ZZZ — die FIXED-LENGTH-Datei wird durch einen **Satz** von Zeichen 1AH beendet (schliesst das Attribut NO-SINGLE-Z ein)

FFZZZ — die Datei wird durch einen Satz abgeschlossen, der mit zwei Zeichen FFH beginnt und mit 1AH aufgefuellt ist

Beispiele:

```
INPUT-ATTRIBUTES = 80,FIXED-LENGTH,NO-SINGLE-Z
INPUT = 128,FIX,NOZZ,FFZZZ
I 128 FI NOZ FF              (minimale Kurzform)
```

2.3.1.2. SORT-FILES

Das SORT-FILES-Kommando spezifiziert die Namen der zu sortierenden Dateien und wahlweise eine Anfangs- und End-Satznummer je Datei.

```
SORT-FILES (=)dateiname[(anfangssatz,endsatz)],...
           (+)
```

= — noch kein SORT-FILES spezifiziert bzw. eingegebenes SORT-FILES-Kommando soll ersetzt werden

+	die angegebene(n) Datei(en) ergaenzen die schon vorher spezifizierte(n); bis zu 32 Eingabedateien koennen in einem Lauf sortiert werden
dateiname	SCP-Dateiname mit wahlweiser Laufwerks- und Typangabe
anfangssatz	eine Zahl von 1 bis 65535 als Nummer des ersten Satzes einer Datei, der in die Sortierung einbezogen werden soll; 0 oder 1 bedeuten Dateibeginn
endsatz	Nummer des letzten zu sortierenden Satzes einer Datei; 0 oder 65535 bedeuten bis Dateiende

Beispiele:

```
SORT-FILE =EIN.DAT
SORT-FILE =B:EIN.DAT(10,120)
SORT-FILES=E1.DAT,E2.DAT,E3.TXT
SORT = B:E1.DAT(1,119),C:E2.DAT(120,65535)
SORT-FILES + E4.DAT,E5.DS1(50,255)
S EIN.DAT                                  (minimale Kurzform)
```

2.3.1.3. MERGE-FILES

Das MERGE-FILES-Kommando spezifiziert die Namen der zu mischenden Dateien.

```
MERGE-FILES(=)dateiname,dateiname,...
           (+)
```

dateiname	SCP-Dateiname
=	ersetzt vorher spezifizierte Misch-Dateinamen durch in diesem Kommando angegebene
+	die in diesem Kommando spezifizierten Dateien ergaenzen die vorher angegebenen; maximal 32 zu mischende Dateien sind moeglich

Beispiele:

```
MERGE-FILE = STAMM.DAT
M STAMM-DAT                (minimale Kurzform)
MERGE + A:STAMM1.DAT,B:STAMM2.DAT
```

Wird ein SORT-FILES- oder ein MERGE-FILES-Kommando eingegeben, prueft SORT 1520 (SCPX), ob die angegebenen Dateien vorhanden

sind und gibt gegebenenfalls eine Fehleranzeige. Das betreffende Kommando kann dann korrigiert werden.

2.3.2. Ausgabedatei

Im OUTPUT-FILE-Kommando wird der Name der Ausgabedatei angegeben, wahlweise ergaenzt durch Spezifikationen.

```
OUTPUT-FILE = dateiname[/C][,satzlaenge] {FIXED-LENGTH}
                                         {CR-DELIMITED}
                                         {VARIABLE    }
                                         {RELATIVE    }
                                         {K-OUTPUT    }
                                         {R-OUTPUT    }
                                         {P-OUTPUT    }
                                         {KR-OUTPUT   }
                                         {KP-OUTPUT   }
                                         {FFZZZ       }
                                         {NO-FFZZZ    }
```

dateiname	Name der Ausgabedatei; muss sich von den Namen aller angegebenen Misch-Eingabedateien unterscheiden, kann aber gleich dem Namen einer Sortier-Eingabedatei sein (letzteres nur in Ausnahmefaellen anwenden, da Eingabedatei ueberschrieben wird!)
/C	Diskette im angegebenen Laufwerk fuer die Ausgabedatei soll vor dem Schreiben der Ausgabedatei gewechselt werden. Durch SORT 1520 (SCPX) erfolgt eine Aufforderung zum Wechseln der Diskette, die nach erfolgtem Austausch mit <ENTER> zu beantworten ist.

Soll die Ausgabedatei vom Typ und der Spezifikation der Eingabedateien (INPUT-FILE-Kommando) nicht abweichen, koennen weitere OUTPUT-FILE-Spezifikationen weggelassen werden.

satzlaenge	1 - 4096, spezifiziert die Satzlaenge in Bytes bzw. die maximale Satzlaenge fuer CR-DELIMITED- oder VARIABLE-Saetze. Angabe wird ignoriert, wenn eine -OUTPUT-Option (K-OUTPUT usw.) spezifiziert ist, da diese Formen eine ganz bestimmte Laenge besitzen (s. Anhang A).
FIXED-LENGTH CR-DELIMITED VARIABLE RELATIVE	Bedeutung wie entsprechende INPUT-Attribute

Die folgenden Bemerkungen sind nur wichtig, wenn der Typ der Ausgabedatei und/oder die Satzlaenge unterschiedlich zum Typ bzw. der Satzlaenge der Eingabedatei angegeben wird, d.h. wenn

SORT 1520 (SCPX) eine Dateikonvertierung durchfuehren soll:

1. Weicht der Typ der Ausgabedatei von dem der Eingabedatei(en) ab und wird die (maximale) Satzlaenge nicht angegeben, erfolgt standardmaessig:

   a. ist die Eingabedatei VARIABLE, aber nicht die Ausgabedatei, so ist die Satzlaenge der Ausgabedatei um 2 Byte kleiner als die der Eingabedatei, da die zwei Laengenbytes geloescht werden

   b. ist die Ausgabedatei VARIABLE, aber nicht die Eingabedatei, so ist die Satzlaenge der Ausgabedatei um 2 Byte groesser als die der Eingabedatei, aber nicht groesser als 4096.

2. Ist die angegebene Satzlaenge der Ausgabedatei kleiner als die Satzlaenge der Eingabedatei, wird der Satz nach rechts abgeschnitten; die Laengenbytes bei VARIABLE werden entsprechend veraendert.

3. Ist die Ausgabedatei FIXED oder RELATIVE, so werden gegebenenfalls die Saetze mit Leerzeichen (20H) bis zur spezifizierten Laenge aufgefuellt.

Die folgenden OUTPUT-Options spezifizieren Ausgabedateien:

K-OUTPUT	Ausgabedatei enthaelt nur die Schluessel als Teil der Saetze der Eingabedatei(en), die Ausgabe-Satzlaenge entspricht der Schluessellaenge
R-OUTPUT	Ausgabedatei enthaelt nur die sortierten Satznummern; ist die Ausgabedatei vom Typ CR-DELIMITED, so werden die Satznummern als ASCII-Text mit je 6 Ziffern angegeben, andere Dateitypen erzeugen 3-Byte-Binaernummern
P-OUTPUT	Ausgabedatei enthaelt nur Zeiger zu den Eingabesaetzen; jeder Zeiger besteht aus der Nummer des Sektors, in dem der Satz beginnt und der relativen Position (Byte-Offset) des Satzbeginns zum Beginn des Sektors; die Zahlenangaben sind entweder ASCII-Zeichen, durch ein Komma getrennt (bei CR-DELIMITED-Dateien) oder binaer in zwei Bytes fuer jede Zahl (andere Datei-Typen)
KR-OUTPUT	Ausgabedatei enthaelt Schluessel und Satznummern
KP-OUTPUT	Ausgabedatei enthaelt Zeiger und Satznummmern

RECORD-NUMBERS — aequivalent zu R-OUTPUT, minimale Kurzform ist REC

KEYS-AND-NUMBERS — aequivalent zu KR-OUTPUT, minimale Kurzform ist KEY-A

Wenn eine der obigen OUTPUT-Options verwendet wird, ausser K-OUTPUT, ist nur eine Eingabedatei erlaubt.

Die folgenden Attribute sind nur fuer Dateien vom Typ FIXED-LENGTH mit binaeren Daten signifikant:

FFZZZ — beendet die Ausgabedatei durch Auffuellen des freien Bereiches im letzten Sektor mit 2 Bytes FFH und nachfolgenden Zeichen 1AH (CTRL Z)

NO-FFZZZ — beendet die Ausgabedatei durch Auffuellen des letzten Sektors der Datei mit 1AH; Standard, wenn nicht FFZZZ fuer die Eingabe spezifiziert ist

Beispiele:

```
OUTPUT-FILE = IN.SUM,PIXED-LENGTH,KR-OUTPUT
OUTPUT = AUS.STM/C,2000,FIX,NO-FFZ
```

2.3.3. Schluessel

Das KEY-Kommando spezifiziert einen oder mehrere Sortierschluessel. Es koennen bis zu 32 Schluessel angegeben werden, jeder mit eigenen Attributen. Mehrfachschluessel koennen in einem einfachen Kommando (wenn noetig, durch "&" Fortsetzung auf der naechsten Zeile) oder in mehreren Kommandos (mit KEY+) spezifiziert werden.

```
KEY (=) (beg,end)                    {ASCENDING       },...
    (+) (#feld-nr,max-l)             {DESCENDING      }
                                     {NUMERIC-ASCII   }
                                     {UPPER-CASE      }
                                     {RIGHT-JUSTIFY   }
                                     {LOHI            }
                                     {MASK-PARITY-BIT }
                                     {ALTSEQ          }
                                     {TWOS-COMPLEMENT }
                                     {COMPUTATIONAL   }
                                     {INTEGER         }
                                     {FLOATING-POINT  }
                                     {PACKED-BCD      }
                                     {COMPUTATIONAL-3 }
```

Ersetzen vorher spezifizierter Schluessel

+	Hinzufuegen zu vorher spezifizierten Schluesseln

Fuer ein Positionsfeld werden zwei Zahlen angegeben:

beg	erste Position des Feldes
end	letzte (enthaltene) Position

Das erste Zeichen eines Satzes hat die Position 1 (bei Saetzen vom Typ VARIABLE befindet sich das erste Datenzeichen auf Position 3).

Bei einer Satzstruktur, bei der die einzelnen Felder durch Komma getrennt sind, werden zur Bezeichnung eines Feldes (Kommafeld) das Zeichen "#" und zwei Zahlen angegeben:

feld-nr	Feldnummer, z.B. Feld #3 befindet sich im Satz zwischen dem zweiten und dritten Komma
max-l	maximale Anzahl der vor dem Feld zu beruecksichtigenden Zeichen, wobei Anfuehrungszeichen, fuehrende und nach rechts auffuellende Leerzeichen nicht gezaehlt werden; ueber max-l hinausgehende Zeichen werden ignoriert; besitzt das Feld weniger Zeichen, so erfolgt ein Auffuellen mit Leerzeichen

Die folgenden Angaben spezifizieren die Sortierreihenfolge, ASCENDING ist Standard:

ASCENDING	Feld aufsteigend sortieren
DESCENDING	absteigend sortieren

Beispiele:

```
KEY = 2,10 ASCENDING     (Bytes 2 bis 10)
KEY = #2,10 ASCENDING    (2. Feld, maximal 10 Zeichen)
KEY = #2,10 DESCENDING   (wie oben, aber absteigend)
K #2 10 DE               (Minimalform)
```

Mehrere Schluessel koennen in einem Kommando spezifiziert werden:

```
KEY = 2,10,ASC, 20,H5,DESC
KEY = #11,40,ASC, #3,5,DESC, #6,8,ASC
KEY = 1,10 ASC, #4,20 DESC
```

Im KEY-Kommando angebbare Testattribute sind:

NUMERIC-ASCII	interpretiert das Feld als Text (im Gegensatz zu binaer), bestehend aus numerischen Zeichen
UPPER-CASE	interpretiert im Feld enthaltene Kleinbuchstaben als Grossbuchstaben
RIGHT-JUSTIFY	fuellt den Feldanfang mit Leerzeichen bei Kommandofeldern, deren Laenge kleiner ist als die spezifizierte maximale Laenge (max-1)
LOHI	testet dieses Feld von rechts nach links, vorgesehen fuer Binaerdaten, bei denen der LOW-Teil zuerst gespeichert ist, kann aber bei jedem Positionsfeld verwendet werden; bei Kommafeldern nicht erlaubt
MASK-PARITY-BIT	hoechstwertiges Bit eines jeden Bytes wird ignoriert
ALTSEQ	sortiert das Feld entsprechend der im COLLATING-SEQUENCE-Kommando spezifizierten Rangfolge (s.u.)

weitere Beispiele:

```
KEY = #17,10,NUMERIC-ASCII,DESCENDING
KEY + #5,10 DESC,NUM-ASC, #8,20,ASCEND,RIGHT,UPPER
KEY = 1,10 LOHI,DESCEND, 11,20 ALTSEQ,ASCEND
```

Die folgenden Testattribute beziehen sich auf nicht-ASCII-Datentypen:

TWOS-COMPLEMENT COMPUTATIONAL	jedes dieser aequivalenten Attribute bedeutet, dass die Daten binaere Festkommazahlen sind, wobei negative Zahlen als Zweierkomplement gespeichert sind. Es wird angenommen, dass der hoeherwertige Teil der Zahl zuerst gespeichert ist (2-Byte-Felder).
INTEGER	binaere Festkommazahl; niederwertiger Teil zuerst gespeichert (2-Byte-Felder)
FLOATING-POINT	Daten im binaeren Gleitkomma-Format; (4-Byte-Felder:einfache Genauigkeit oder 8-Byte-Felder:doppelte Genauigkeit)
PACKED-BCD COMPUTATIONAL-3	jedes dieser aequivalenten Attribute bezeichnet gepackte binaercodierte Dezimalzahlen (BCD) mit oder ohne Vorzeichen;

um Fehlinterpretationen zu vermeiden, sollte COMPUTATIONAL-3 nur mit COMP-3 oder C3 abgekuerzt werden

Fuer vorzeichenlose binaere Festkommazahlen (hoechstwertiges Bit repraesentiert Ziffernstelle), die in der Reihenfolge L-Teil, H-Teil gespeichert sind, sollte LOHI benutzt werden, bei Speicherung H-Teil, L-Teil wird kein Attribut benoetigt.

weitere Beispiele:

```
KEY = 1,1 TWOS-COMPLEMENT,3,4,TWOS-COMPLEMENT
KEY = 1,2 INTEGER,DESC,3,6 FLOATING-POINT,11,14 FLOAT
KEY + 11,12 TWOS-COMPLEMENT,5,10 PACKED-BCD
KEY  + 11,12 COMPUTATIONAL,5,10 COMP-3 (aequivalent vorhergehender Zeile)
```

2.3.4. Start

Sind alle gewuenschten Kommandos eingegeben, wird die Ausfuehrung durch die Eingabe des Kommandos

GO

gestartet.

Das GO-Kommando (oder einfach G) bringt bei unvollstaendiger Kommandoeingabe (z.B. keine INPUT-ATTRIBUTES spezifiziert) eine entsprechende Fehlerausschrift (siehe Anhang B). Eine Fehlerausschrift erfolgt auch bei sich widersprechenden Spezifikationen (z.B. Schluesselposition groesser als definierte Eingabesatzlaenge).

Nach einer Fehlerausschrift sind die vorher eingegebenen Kommandos noch gueltig, so dass man nur die betreffenden fehlerhaften Kommandos zu aendern hat und mit GO erneut starten kann.

Nachfolgend einige Beispiele fuer komplette Kommandofolgen:

```
INPUT-ATTRIBUTES = 62,CR-DELIMITED
SORT-FILE = EIN.DAT
OUTPUT-FILE = AUS.DAT
KEY = #2,4 ASCENDING RIGHT-JUSTIFY
GO
```

```
INPUT = 80,FIXED
SORT LIST.NEU (1,1000)
MERGE LIST.STA
KEY = 64,70 ASC,31-40 ASC,1,19 ASC UPPER
OUTPUT = LIST.STA/C
GO
```

```
INPUT = 1024,CR-DEL
MERGE LAG1.ART,LAG2.ART,LAG3.ART
KEY = #3,10,NUMERIC-ASCII,DESC
OUTPUT = LAGER.ART,1000,FIXED
GO

I = 64,CR-D;S=EIN.DAT;O=AUS;K=#2,8,DE;G
```

2.4. Kommandodatei

Einige oder alle Kommandos koennen zu einer Kommandodatei zusammengefasst werden, dadurch vereinfacht sich die Kommandoeingabe bei wiederholt durchzufuehrenden gleichartigen Sortier- und/oder Mischvorgaengen wesentlich.

Die Kommandos in dieser Datei haben genau das gleiche Aussehen wie bei einer einzelnen Eingabe an der Konsole. Die Datei kann mit einem Texteditor (z.B. TP 1520 (SCPX)) angelegt werden.

Die Kommandodatei wird mit dem Kommando

CFILE=dateiname

aufgerufen.

Beispiele:

```
CFILE = SORT.TXT
CF B:UNSORT.STA
```

Eine Kommandodatei muss nicht die komplette Kommandofolge enthalten.

Kommandos koennen vor dem Aufruf der Kommandodatei eingegeben werden und nach dem CFILE-Kommando, wenn die Kommandodatei ein GO enthaelt.

Zur Anzeige der in der Kommandodatei enthaltenen Kommandos dient das Kommando

LIST

, das entweder Bestandteil der Kommandodatei sein kann oder mittels Tastatur vor Aufruf der Kommandodatei eingegeben wird. Die Anzeige beginnt mit der nach LIST folgenden Kommandozeile.

2.5. Arbeitsdatei

Fuer alle Sortierungen, ausser bei sehr kleinem Datenbestand, benutzt SORT 1520 (SCPX) eine Arbeitsdatei auf Diskette. Standardmaessig befindet sich diese Datei auf der im aktuellen Laufwerk vorhandenen Diskette.

Diese Standardzuordnung kann durch das Kommando:

WORK-DRIVE = laufwerk

veraendert werden. Dabei ist "laufwerk" ein Buchstabe A bis E entsprechend der Systemkonfiguration, wahlweise gefolgt von einem Doppelpunkt ":".

Beispiele:

```
WORK-DRIVE = C:
WORK E
```

Bei umfangreichen zu sortierenden Datenmengen sollte ein entsprechend freier Bereich auf einer Diskette reserviert (etwa in Groesse der Ausgabedatei) und das WORK-DRIVE-Kommando benutzt werden.

Die Arbeitsdatei erhaelt den Namen SORT.□□□ und wird nach erfolgter Sortierung automatisch geloescht.

2.6. TAGSORT

Das Kommando

TAGSORT

veranlasst SORT 1520 (SCPX), einen speziellen Algorithmus zum Erzeugen der Ausgabedatei anzuwenden.

Dieses Verfahren bewirkt eine Reduzierung bzw. einen Wegfall des fuer die Arbeitsdatei benoetigten Diskettenbereiches (2-Laufwerksanlagen).

TAGSORT darf nur verwendet werden, wenn nur eine zu sortierende Eingabedatei und keine zu mischenden Eingabedateien vorliegen.

2.7. Ergebnisausschrift

SORT 1520 (SCPX) besitzt sechs Varianten der Ausschrift nach Abschluss des Sortier- bzw. Mischvorganges. Die Auswahl der gewuenschten Variante erfolgt durch das Kommando

PRINT-LEVEL = n

wobei n eine Zahl von 0 bis 5 repraesentiert (2 ist Standard).

Beispiele:

```
PRINT-LEVEL = 4
PR 1
```

- Variante 0: keine Anzeige

- Variante 1:
  .Anzahl der sortierten Saetze, wenn ≠ 0
  .Anzahl der gemischten Saetze, wenn ≠ 0
  .Anzahl der Ausgabesaetze, wenn Angabe nicht redundant
  .Groesse der Arbeitsdatei in KByte, wenn ≠ 0

- Variante 2:
  (Standard)
  wie Variante 1, plus:
  .Anzahl der SORT-Eingabesaetze, wenn nicht redundant
  .Anzahl der MERGE-Eingabesaetze, wenn nicht redundant
  .Anzahl der bei Selektion nicht behandelten Saetze:
  der SORT-Eingabesaetze, wenn ≠ 0
  der MERGE-Eingabesaetze, wenn ≠ 0

- Variante 3:
  wie Variante 2, plus:
  .Anzahl der Sortierlaeufe
  .Anzahl der Mischlaeufe

- Variante 4:
  wie Variante 3, plus Text:
  SORTING...
  MERGING...
  ADDITIONAL MERGE...
  FINAL MERGE...

- Variante 5:
  wie Variante 4, plus:
  .Anzahl der Bytes des Arbeitsspeichers
  .Anzahl der Eingabesaetze bei jedem Sortierlauf
  .Anzahl der sortierten Eingabebloecke bei jedem Mischlauf

2.8. Weitere Kommandos

CANCEL — loescht alle vorher eingegebenen Kommandos (ausser COLLATING-SEQUENCE-Kommandos)

BYE — verlaesst SORT, aequivalent zu CTRL C (Warmstart)

NO-ERROR-MESSAGES — tritt ein Fehler waehrend der Abarbeitung auf, so wird nur die Fehlernummer, nicht der gesamte Text angezeigt; der fuer den Fehlertext benoetigte Speicherbereich ist zusaetzlich als Arbeitsspeicher verfuegbar;
Kurzform: NO-E

ERROR-MESSAGE zahl	zeigt fuer den Fehler mit der Nummer "zahl" den gesamten Fehlertext an; Angabe nur nach einem NO-ERROR-MESSAGE-Kommando sinnvoll
RETURN-TO-CONSOLE	kehrt nach Abarbeitung nicht zum Betriebssystem zurueck, sondern ermoeglicht die Eingabe von Kommandos fuer einen naechsten Sortier-/Mischlauf; Kommando reduziert den verfuegbaren Arbeitsspeicherbereich; ist nicht zu empfehlen bei grossen Datenmengen oder maximaler Verarbeitungsgeschwindigkeit

2.9. Selektion

Es besteht die Moeglichkeit, nur mit ausgewaehlten Saetzen einen Sortier- bzw. Mischvorgang durchzufuehren.

2.9.1. Einfuehrung

SELECT und EXCLUDE definieren Kriterien fuer die Selektion von Saetzen auf der Basis von Datentests. Bei SELECT werden alle die Testkriterien erfuellenden Saetze in den Sortier-/Mischprozess einbezogen. Bei EXCLUDE dagegen werden in diesen Prozess alle die Saetze einbezogen, bei denen die Testkriterien nicht erfuellt sind.

SELECT und EXCLUDE koennen bei Sortier- und Mischvorgaengen verwendet werden.

Die SELECT- und EXCLUDE-Kommandos haben eine gemeinsame Syntax mit vielen Optionen und Varianten. Die gebraeuchlichste Form ist:

```
(SELECT) (=) (FIELD beg,end) operator"text" {NUMERIC-ASCII   }
(EXCLUDE)(+) ([FIELD]#feld-nr)              {UPPER-CASE      }
                                            {RIGHT-JUSTIFY   }
                                            {LOHI            }
                                            {MASK-PARITY-BIT}
                                            {ALTSEQ          }
```

=	ersetzen ein vorher eingegebenes SELECT/EXCLUDE-Kommando bzw. Ersteingabe
+	ergaenzen vorher eingegebene gueltige SELECT/EXCLUDE-Kommandos
FIELD beg,end	beg und end bezeichnen ein Positionsfeld durch Angabe der Nummer des ersten und letzten Bytes, wie beim KEY-Kommando

FIELD #feld-nr	bezeichnet ein variables Kommafeld, wie beim KEY-Kommando, ausser dass hier keine Maximallaenge angegeben wird; das vollstaendige Feld wird in den Test einbezogen
operator	einer der nachfolgenden Vergleichsoperatoren: < oder LT kleiner < = LE kleiner oder gleich = EQ gleich <> NE ungleich > = GE groesser oder gleich > GT groesser
"text"	ist der Text, mit dem das spezifizierte Feld verglichen wird, eingeschlossen in Anfuehrungsstrichen
NUMERIC-ASCII UPPER-CASE RIGHT-JUSTIFY LOHI MASK-PARITY-BIT ALTSEQ	analoge Bedeutung wie beim KEY-Kommando; beziehen sich auf beide, Feld und "text"

Beispiele:

```
SELECT = FIELD #2>="841014"
SELECT = FIELD #3="UEBWS"
SE #3 = "UEBWS"              (aequivalent vorhergehender Zeile)
SELECT = FIELD #7>"300>NUMERIC-ASCII
SELECT + #6LE"400"NUM-ASC
```

Werden Mehrfach-SELECT- und/oder EXCLUDE-Kommandos angegeben (s.o., letzte beide Zeilen), so werden all jene Saetze in das Sortieren/Mischen einbezogen, die bei SELECT alle Be-dingungen erfuellen bzw. bei EXCLUDE nicht erfuellen.

2.9.2. Bereichstest

Der Test, ob der Wert eines Feldes in einem bestimmten Bereich liegt, kann auch mit den Operatoren "between" und "not-between" erfolgen.

(SELECT)(=) Feldangabe wie oben (BT) "text1","text2" Attribute wie oben
(EXCLUDE)(+) (NB)

BT	zwischen "text1" und "text2" (einschliesslich)
NB	nicht zwischen "text1" und "text2"
"text1"	untere Bereichsgrenze

"text2" obere Bereichsgrenze

Beispiele:

```
SELECT #2 NB"-200","200" NUMERIC-ASCII
SELECT = FI 33,41 BT "100","999"
```

2.9.3. Weitere Selektionsmoeglichkeiten

- Vertauschen von Feldern und Konstanten:
  Angabe der Konstanten vor und des Feldes nach dem Operator; Vergleich eines Feldes mit einem oder mehreren Feldern im gleichen Satz:

  ```
  SELECT = "UEBWS" = FIELD #3
  SELECT = FIELD #1 > FIELD #2
  SELECT = "128" BT FIELD #1, FIELD #2
  ```

  Der letzte Test ist erfuellt, wenn Feld 1 kleiner oder gleich 128 und Feld 2 groesser oder gleich 128 ist.

- Logische Operatoren:
  komplexe Bedingungen koennen durch Verwenden der logischen Operatoren AND, OR und XOR angegeben werden:

  ```
  SELECT = #2 > "200"NUM OR #1 = "SUM"
  SELECT = #2 GT "200"NUM AND NOT #1 = "SUM"
  ```

  AND, OR, XOR und NOT koennen nicht abgekuerzt werden.

- Klammerung:
  Klammern koennen zur Zusammenfassung logischer Operationen verwendet werden; ohne Klammerung erfolgt die Verarbeitung in der Reihenfolge: alle Vergleiche zuerst, dann NOT, AND, XOR und OR:

  ```
  SELECT = #3 = "A" AND (#4 >="615" OR #1 = "XY")
  ```

2.9.4. Selektion bei nicht-ASCII-Daten

Eine Selektion von Saetzen mit BCD-Daten und verschiedenen Typen von Binaerdaten ist durch die Angabe spezieller Testattribute und Konstantendarstellungen ebenfalls moeglich.

2.9.4.1. Spezielle Testattribute

Die folgenden Testattribute koennen ebenfalls zur Selektion von Saetzen benutzt werden. Ihre Bedeutung ist wie die der entsprechenden KEY-Attribute.

```
TWOS=COMPLEMENT oder COMPUTATIONAL
INTEGER
FLOATING POINT
PACKED-BCD oder COMPUTATIONAL-3
```

Zuerst einige Beispiele zur Anwendung obiger Testattribute beim Vergleich von Feldern:

```
SELECT = FIELD 1,2 > FIELD 3,4 INTEGER
SELECT = FIELD 1,2 < FIELD 3,4 LOHI
SELECT = FIELD 1,8 BT FIELD 9,16,FIELD 17,24 FLOATING
SELECT = FIELD 1,10 < 21,30 COMPUTATIONAL-3
```

2.9.4.2. Numerische Konstanten

Die gegeneinander zu testenden Werte koennen statt eines in Anfuehrungszeichen eingeschlossenen Textes auch numerische Konstanten sein. Die numerischen Konstanten repraesentieren die im Buerocomputer gespeicherten binaeren Werte, ausgedrueckt als Oktal-, Dezimal-, Hexadezimal- oder BCD-Zahlen. Es kann die Reihenfolge der internen Speicherung angegeben werden, ob der hoeher- oder niederwertige Teil zuerst gespeichert ist.

Numerische Konstanten werden verwendet, wenn Binaer- oder BCD-Daten getestet werden, waehrend der Text bei ASCII-Daten genutzt wird. (Numerische Konstanten koennen auch bei ASCII-Daten verwendet werden, dazu muss die ASCII-Codierung eines jeden Zeichens eingegeben werden.)

SORT 1520 (SCPX) erlaubt die explizite Angabe der Konstantenlaenge in Bytes. Enthaelt die Konstante weniger signifikante Ziffern, so werden die fehlenden hoeherwertigen Bytes durch SORT hinzugefuegt, bei negativen Zahlen (ausser bei BCD-Konstanten) Bits mit dem Wert 1, in allen anderen Faellen Bits mit dem Wert 0.

Eine numerische Konstante besteht aus einem optionalen Vorzeichen, einer Anzahl von Ziffern, die in einer Zeile eingegeben werden koennen. Optional ist die Angabe der Zahlendarstellung und, wenn eine Angabe der Zeichendarstellung erfolgt, optional eine Dezimalzahl, die die Laenge der Konstanten in Bytes darstellt. Vor dem Buchstaben, der die Zahlendarstellung spezifiziert bzw. zwischen diesem und einer nachfolgenden Laengenangabe, darf kein Leerzeichen stehen.

Die Zahlendarstellung wird wie folgt definiert:

T dezimal, Zahl gewandelt in Binaerzahl

I dezimal, gewandelt in Binaerzahl, niederwertiger Teil zuerst gespeichert

P dezimal, gewandelt in BCD-Zahl

H hexadezimal

Q oktal

Wird keine Zahlendarstellung spezifiziert, wird dezimal (T) angenommen, ausser, wenn die Konstante mindestens ein Zeichen A - F enthaelt, dann hexadezimal.

Alle numerischen Konstanten werden von werthoechster (an adressenniederer Stelle) zu wertniedrigster Stelle im Speicher abgelegt, ausser denen, die als I-Konstanten definiert wurden, wo das niederwertige Byte des Feldes ebenfalls zuerst gespeichert ist, d.h. wo das LOHI- oder INTEGER-Testattribut verwendet wird. Wird eine hexadezimale Konstante bei einem solchen Test verwendet, dann muss sie so angegeben werden, dass das niederwertige Byte zuerst geschrieben wird.

Beispiele:

Konstante	Byteanzahl	Speicherdarstellung (hexadezimal)
12	1	0C
12T	1	0C
12T2	2	00 0C
12I2	2	0C 00
12H	1	12
12P	1	12
-2	1	FE
-2I4	4	FE FF FF FF
1234P	2	12 34
Konstante	Byteanzahl	Speicherdarstellung (hexadezimal)
+1234P	3	01 23 4C
-1234P4	4	00 01 23 4D
1029AH	3	01 02 9A
1029A	3	01 02 9A (als hexa angenommen)
0001H	1	01 (fuehrende Nullen nicht abgespeichert)
1H2	2	00 01

Einige Beispiele fuer die Verwendung numerischer Konstanten in SELECT-Kommandos:

```
SELECT = FIELD 17,17 = 4AH
Test auf 4A hexa in Position 17

SELECT = FIELD 1,8 < 1024P8 COMPUTATIONAL-3
```

nimmt Saetze, bei denen in Positionen 1-8, interpretiert als gepackte Dezimalzahl, ein Wert kleiner als 1024 enthalten ist.

2.9.4.3. Konstantenliste

In Ergaenzung zu Feldern und einfachen Konstanten kann noch eine andere Form von Werten angegeben werden: die Konstantenliste. Eine Konstantenliste ist eine Folge von Konstanten, getrennt durch Komma oder Leerzeichen und eingeschlossen in eckige Klammern. Die Konstanten brauchen nicht von gleichem

Typ zu sein. Ein Teil kann als Text und ein anderer als numerische Konstante angegeben werden:

```
SELECT = FIELD 1,6 = ["SUMM",0D0AH]
```

2.9.5. Veraenderung der Rangfolge

Mit dem COLLATING-SEQUENCE-Kommando kann die Rangfolge der Zeichen spezifiziert oder modifiziert werden. Diese veraenderte Rangfolge wird beim Feldtest im KEY-, SELECT- und EXCLUDE-Kommando durch das Attribut ALTSEQ wirksam.

```
COLLATING-SEQUENCE (=) argument,argument,...
                   (+)
```

(+ und = besitzen gleiche Wirkung).

Standardmaessig ist die ASCII-Zeichenfolge (8-Bit-Binaerwertfolge) als Rangfolge definiert, d.h. auf Position 0 befindet sich das Zeichen mit dem Binaerwert 00H oder auf Position 49 die Ziffer "1" als Zeichen mit dem Binaerwert 31H bzw. auf Position 65 das ASCII-Zeichen "A" mit dem Binaerwert 41H.

Damit gilt beim Vergleich "1" < "A".
Eine Veraenderung dieser Rangfolge durch das COLLATING-SEQUENCE-Kommando kann in zwei Formen erfolgen: implizit und explizit.

- Implizite Positionsargumente

  Die Position ergibt sich aus der Reihenfolge der angegebenen Zeichen, die allgemein in Anfuehrungszeichen eingeschlossen oder als Code dezimal, hexadezimal oder oktal dargestellt werden koennen.

  Beispiele:

  ```
  COL = " ", "A-Z", "0-9", 7FH
  ```

  Das erste Zeichen, hier das Leerzeichen, wird stets der Position 0 zugeordnet, das zweite Zeichen der Position 1 usw.
  Im angefuehrten Beispiel ist das zweite Zeichen der Beginn eines Bereiches von Zeichen, naemlich der Buchstaben "A bis Z". Bei der Angabe von Bereichen werden erstes und letztes Zeichen durch einen Bindestrich getrennt. Nicht angefuehrte Zeichen behalten ihre Standardposition.

- Explizite Positionsargumente

  Eine Position wird durch eine numerische Konstante oder ein in Anfuehrungszeichen eingeschlossenes Zeichen direkt angegeben, gefolgt von einem "+" fuer steigende, einem "-" fuer fallende oder keinem Zeichen fuer gleiche Positionen.

Beispiele:

a) COL = "a" - "z" = 65+, "A" - "Z" = 65+
   d.h. "a" = "A" (Position 65)
   "b" = "B" (Position 66)
   Klein- und Grossbuchstaben besitzen gleiche Positionen.

b) COL = "0" - "9" = 50-
   ist gleich
   COL = "9" - "0" = 40+

c) COL = "0" - "9" = 50
   alle Ziffern haben die gleiche Position

- Gemischte Angaben

Folgt einem expliziten Positionsargument ein implizites Argument, so wird ihm die Position des letzten expliziten Argumentes +1 zugeordnet, unabhaengig davon, ob die vorhergehenden Argumente steigend, fallend oder von konstanten Positionen sind.
Ein Argument setzt ein links von ihm stehendes Argument ausser Kraft.

Beispiel:

COL 0 - 255 = 255, "." = 0, "0" - "9"

alle Zeichen (0-255) erhalten explizit die Position 255, implizit der Punkt die Position 0 und die Ziffern "0 bis 9" die Positionen 1 - 10.

- Mehrfach-Kommandos

Werden mehrere COLLATING-SEQUENCE-Kommandos angegeben, so beginnt das zuletzt eingegebene Kommando implizit bei Position 0, d.h. ein "+" nach dem Kommandonamen hat ergaenzende Wirkung wie bei anderen Kommandos (z.B. KEY+...).

- Reinitialisieren

Es erfolgt kein automatisches Ruecksetzen einer veraenderten Rangfolge zur Standardrangfolge durch das CANCEL-Kommando. Durch

COL 0 - 255

wird aber die Standardrangfolge wiederhergestellt.

2.10. ANHANG A: Formate

Uebersicht

1. Dateiformate
1.1. CR-DELIMITED
1.2. FIXED-LENGTH
1.3. VARIABLE
1.4. RELATIVE

2. Feldformate
2.1. Positionsfelder
2.2. Kommafelder

3. Datenformate
3.1. NUMERIC-ASCII
3.2. PACKED-BCD (COMPUTATIONAL-3)

4. Ausgabedateiformate
4.1. K-OUTPUT (nur Schluessel)
4.2. R-OUTPUT (Satznummern)
4.3. P-OUTPUT (Zeiger)
4.4. KR-OUTPUT (Schluessel und Satznummern)
4.5. KP-OUTPUT (Schlussel und Zeiger)

1. Dateiformate

SORT 1520 (SCPX) kann Dateien behandeln, die Saetze eines der vier Typen CR-DELIMITED, FIXED-LENGTH, VARIABLE oder RELATIVE enthaelt.

1.1. CR-DELIMITED

Eine SCP-Datei besteht normalerweise aus Textzeilen (=Saetze), die durch carriage return (0DH, Wagenrichtung und Zeilenschaltung auf Zeilenanfang) und line feed (0AH, Zeilenschaltung) abgeschlossen werden. Endet der letzte Satz der Datei nicht genau am Sektorende, folgt dem letzten Satz ein CTRL Z (1AH). Endet der letzte Satz genau am Sektorende, so kann abhaengig vom dateierzeugenden Programm oder der verwendeten Sprache ein zusaetzlicher Sektor mit einem CTRL Z die Datei abschliessen.
SORT 1520 (SCPX) erzeugt dieses Format fuer die Ausgabedatei. Fuer Eingabedateien sind zum oben beschriebenen Format Variationen moeglich.

Wird eine CR-DELIMITED-Datei gelesen, so wird auch ein einzelnes Zeichen carriage return oder line feed als Satztrennzeichen akzeptiert. Zusaetzliche carriage return oder line feed vor dem Beginn des naechsten Satzes werden ignoriert.

Eine Datei benoetigt also nur ein Trennzeichen zwischen den Saetzen; Leerzeilen, die nicht mindestens ein Leerzeichen enthalten, werden bei der Eingabe ignoriert.

Ein CTRL Z wird auch als Satztrennzeichen angesehen, so dass der letzte Satz der Datei auch dann korrekt gelesen wird, wenn das carriage return nach ihm fehlt.

SORT 1520 (SCPX) erkennt das Dateiende, wenn das physische Dateiende erreicht ist oder ein CTRL Z gelesen wird. Die Eingabeattribute NO-SINGLE-Z und NO-ZZZ wirken bei CR-DELIMITED-Eingabedateien nicht.

Wird eine CR-DELIMITED-Ausgabedatei erzeugt, so werden ein carriage return und ein line feed nach jedem Satz geschrieben, unabhaengig davon, wie der entsprechende Eingabesatz abgeschlossen war. Die Datei wird abgeschlossen, indem der freie Bereich des letzten Sektors mit CTRL Z's oder zwei Zeichen FFH, gefolgt von CTRL Z's wenn das FFZZZ-Ausgabeattribut spezifiziert wurde, aufgefuellt wird. Reichen die Daten bis zum Sektorende, werden keine EOF-Zeichen ausgegeben.

CR-DELIMITED-Dateien enthalten normalerweise nur ASCII-Textdaten (20H - 7FH), bei anderen Daten muss sicher sein, dass sich keine Codes 1AH, 0DH oder 0AH innerhalb des Satzes befinden.

1.2. FIXED-LENGTH

FIXED-LENGTH-Dateien bestehen aus Saetzen fester Laenge und koennen alle Zeichen ausser 1AH enthalten. Die Satzlaenge ist unabhaengig von der Sektorlaenge. Sie kann 1 - 4096 Bytes betragen.

Bei Eingabedateien wird das Dateiende erkannt, wenn bis zum physischen Dateiende weniger Bytes als die spezifizierte Satzlaenge bleiben. Wurde entweder NO-SINGLE-Z oder NO-ZZZ spezifiziert, dann wird das Ende der Datei durch ein CTRL Z (1AH) als erstes Zeichen eines Satzes erkannt.

Bei Ausgabedateien wird der ungenutzte Bereich des letzten Sektors der Datei entweder mit CTRL Z's (1AH) oder zwei FFH, gefolgt von CTRL Z's aufgefuellt (Attribut FFZZZ). Endet der letzte Satz am Sektorende, werden keine CTRL Z'sausgegeben. FFZZZ wird standardmaessig fuer die Ausgabedatei angenommen, wenn es fuer die Eingabedatei spezifiziert wurde.

1.3. VARIABLE

Dateien vom Satztyp VARIABLE enthalten Saetze variabler Laenge, wobei die Laenge jedes Satzes binaer in den ersten beiden Bytes des Satzes enthalten ist. Im ersten Laengenbyte ist der H-Teil der Satzlaenge gespeichert.
Diese Satzlaenge kann im Bereich von 1 - 4094 liegen, wobei die zwei Laengenbytes nicht einbezogen sind. Es ist zu beachten, dass bei der Spezifikation der Maximalsatzlaenge der Datei diese zwei Bytes einzubeziehen sind. Das erste Datenzeichen befindet sich immer auf Position 3.

Das Dateiende bei Eingabedateien wird erkannt, wenn bis zum physischen Dateiende weniger als zwei Bytes verbleiben oder wenn die Laengenbytes zwei CTRL Z's (1AH) enthalten.

Eine Ausgabedatei wird durch zwei Bytes 00H beendet (Null-Laengensatz), eventuell verbleibender freier Bereich bis zum Sektorende wird mit CTRL Z's aufgefuellt (bzw. 2FFH), gefolgt von 1AH bei spezifiziertem FFZZZ).

VARIABLE-Dateien koennen beliebige Datenzeichen ab Position 3 eines jeden Satzes enthalten.

1.4. RELATIVE

RELATIVE-Dateien sind ein Spezialtyp von Dateien fester Satzlaenge. Die Saetze der Datei sind von 1 ab durchnumeriert anzusehen (Satznummer nicht gespeichert). Geloeschte Saetze bestehen aus Binaernullen (00H) in der spezifizierten Satzlaenge von 1 - 4096 Bytes.

In die Ausgabedatei werden geloeschte Saetze nicht uebernommen. Jede RELATIVE-Datei besitzt am Dateianfang ein "Header"-Feld aus 6 Bytes, das folgende Angaben enthaelt:

Byte	Inhalt
0	0 als Kennzeichen fuer RELATIVE-Datei
1,2	Satzlaenge binaer, L-Teil zuerst gespeichert
3	frei
4,5	hoechste Satznummer der in der Datei enthaltenen Saetze

Der erste Satz beginnt ab Byte 6 des ersten Sektors. Die Saetze koennen beliebige Daten enthalten.

2. Feldformate

Bei KEY, SELECT bzw. EXCLUDE erfolgt die Angabe eines oder mehrerer Felder, die als Positionsfelder oder Kommafelder definiert werden koennen.

2.1. Positionsfelder

Ein Positionsfeld wird gekennzeichnet durch eine Anfangs- und Endposition. Die Positionen (Spalten) beginnen mit 1 beim ersten Byte eines Satzes oder 3 beim ersten Datenbyte einer VARIABLE-Datei. Kommas, Anfuehrungszeichen oder andere Zeichen sind Bestandteil des Satzes und erfahren keine Sonderbehandlung.

Geht die Endeposition eines Feldes ueber das Ende eines CR-DELIMITED-Satzes hinaus, werden den restlichen Positionen Leerzeichen (20H) zugeordnet. Befindet sich das gesamte Feld ausserhalb des Satzes, erfolgt zusaetzlich eine Fehlerausschrift.

2.2. Kommafelder

Ein Kommafeld wird durch seine Feldnummer spezifiziert.
Feld n ist dabei das Feld, das nach dem (n-1)-ten Komma (ausser in Anfuehrungszeichen eingeschlossene Kommas) beginnt. Ist n=1, beginnt das Feld am Satzbeginn (Position 3 bei VARIABLE-Saetzen). Das naechste Komma (ausser in Anfuehrungszeichen eingeschlossene) bzw. das Satzende kennzeichnen das Ende des Feldes.
Leerzeichen (20H) am Beginn oder Ende eines Feldes werden ignoriert, ebenso alle Anfuehrungszeichen im Feld.

Ein Schluesselfeld, das nach dem Eliminieren dieser Zeichen noch groesser ist als die spezifizierte Maximallaenge, wird nach rechts abgeschnitten.

Ein Kommafeld kann bis zu 4096 Zeichen lang sein, die Feldnummer muss im Bereich 1 - 255 liegen.

3. Datenformate

Das Basisdatenformat von SORT 1520 (SCPX) ist ein Strom vorzeichenloser (unsigned) Bytes. Werden zwei Schluessel waehrend des Sortierens oder Mischens verglichen oder ein Satzselektionstest durchgefuehrt, werden die zwei Bytestroeme von links nach rechts verglichen bis eine Position gefunden ist, an der die Bytes ungleich sind. So werden ASCII-Daten und vorzeichenlose binaere Festkommazahlen (unsigned binary fixed point), bei denen der H-Teil (hoeherwertige Teil) zuerst gespeichert ist, sortiert. Dabei gibt es zwischen beiden Typen keinen Unterschied. Nachfolgend werden weitere Datenformate erlaeutert.

3.1. NUMERIC-ASCII

Ein NUMERIC-ASCII-Datum ist eine Zahl, repraesentiert durch eine Folge von ASCII-Zeichen. Eine in untenstehender Reihenfolge angeordnete Zeichenfolge eines Feldes wird dabei als

korrekt interpretiert:

- eine beliebige Anzahl von Leerzeichen
- wahlweise "+" oder "-", gefolgt von einer Anzahl Leerzeichen
- fuehrende Nullen
- Ziffern
- wahlweise Dezimalpunkt
- zusaetzliche Ziffern bei vorhandenem Dezimalpunkt
- Leerzeichen
- E, e, D oder d als Kennzeichen eines nachfolgenden Exponenten
- wenn E, e, D oder d:
  . Leerzeichen
  . wahlweise "+" oder "-", wahlweise gefolgt von Leerzeichen
  . bis zu 3 Ziffern zwischen -255 und +255

Maximal 14 signifikante Ziffern sind zugelassen.
Kann nur ein Teil eines als NUMERIC-ASCII deklarierten Feldes numerisch interpretiert werden, so wird der so erhaltene Wert benutzt. Ist das gesamte Feld nicht numerisch, wird der Wert 0 angenommen.

Ein Exponent ausserhalb des Bereiches von -255 bis +255 oder mehr als 3 Ziffern fuer den Exponenten ergeben einen undefinierten Wert. Eine Fehlerausschrift erfolgt nicht.

3.2. PACKED-BCD (COMPUTATIONAL-3)

Das PACKED-BCD (Binary Coded Decimal)-Attribut definiert gepackte Dezimalzahlen. Bei diesen Daten werden zwei Ziffern in einem Byte gespeichert, die hoeherwertige Ziffer in den hoechsten Bits; die hoeherwertigen Bytes zuerst (an adressniederer Speicherstelle). Jede Ziffer wird in 4 Bits dargestellt, das optionale Vorzeichen befindet sich in den niederen (rechten) 4 Bits des letzten Bytes. Die Werte in diesen 4 Bits besitzen folgende Bedeutung:

0 - 9	Ziffer: Zahl ist ohne Vorzeichen
A, C, E oder F hexa	+ Vorzeichen
B oder D hexa	- Vorzeichen

SORT 1520 (SCPX) vergleicht korrekt PACKED-BCD-Zahlen mit und ohne Vorzeichen miteinander, auch Zahlen unterschiedlicher Vorzeichendarstellung.

4. Ausgabedateiformate

Es gibt verschiedene Ausgabemoeglichkeiten. In allen Faellen beginnen die Saetze mit zwei Bytes Satzlaenge, wenn VARIABLE-Ausgabedatei spezifiziert ist und enden mit carriage return und line feed bei CR-DELIMITED-Ausgabedateien.
Wurde eine der Ausgabemoeglichkeiten spezifiziert, ist eine angegebene Ausgabesatzlaenge ohne Bedeutung. Die Ausgabesatzlaenge fuer die verschiedenen Optionen sind die folgenden:

Dateityp	K-OUTPUT	R-OUTPUT	P-OUTPUT	KR-OUTPUT	KP-OUTPUT
CR-DELIMITED	Keys	6	13	Keys+6	Keys+13
FIXED-LENGTH	Keys	3	4	Keys+3	Keys+4
VARIABLE	Keys+2	5	6	Keys+5	Keys+6
RELATIVE	Keys	3	4	Keys+3	Keys+4

"Keys" bezeichnet die Schluessellaenge, wie unten bei K-OUTPUT beschrieben. Zur Satzlaenge bei CR-DELIMITED-Dateien werden carriage return und line feed nicht mitgezaehlt.

4.1. K-OUTPUT (nur Schluessel)

Jeder "Ausgabesatz" einer K-OUTPUT-Datei enthaelt nur die Schluesselbegriffe. Die Satzlaenge ist die Gesamtlaenge der modifizierten Schluessel. PACKED-BCD-Felder besitzen ein Byte mehr, alle NUMERIC-ASCII-Schluessel haben 9 Bytes.

Die Schluessel sind "verkettet", der Schluessel mit der hoechsten Prioritaet zuerst. Bei Kommafeldern werden, wie oben beschrieben, Leerzeichen und Ausfuehrungszeichen eliminiert; sie werden bis zu ihrer Maximallaenge mit Leerzeichen aufgefuellt. Spezifizierte Attribute werden wie folgt verwendet:

NUMERIC-ASCII	Daten in 9-Byte-Format konvertiert (siehe unten)
UPPER-CASE	Kleinbuchstaben in Grossbuchstaben umgewandelt
RIGHT-JUSTIFY	fuehrende Leerzeichen werden bei Kommafeldern bis zum Erreichen der Maximallaenge hinzugefuegt
LOHI	umgekehrte Bytereihenfolge
MASK-PARITY-BIT	in allen Bytes Bit 7 gleich 0 gesetzt
ALTSEQ	Position jedes Bytes entsprechend COLLATING-SEQUENCE-Kommando veraendert
TWOS-COMPLEMENT	Zweierkomplement (Komplement Bit 7 des ersten Bytes)

INTEGER	umgekehrte Bytereihenfolge und Komplement Bit 7 des ersten Bytes
FLOATING-POINT	siehe unten
PACKED-BCD	ein Byte FFH wird dem Schluessel vorangestellt; ist ein Vorzeichen vorhanden, so wird dies durch eine Rechtsverschiebung um 4 Bits geloescht; war das Vorzeichen negativ, so wird das Einerkomplement des gesamten Schluessels, einschliesslich des hinzugefuegten Bytes, gebildet
DESCENDING	Einerkomplement des gesamten Schluessels, nachdem alle anderen Attribute beruecksichtigt wurden

Eine K-OUTPUT-Datei, die ohne Angabe von Attributen gebildet wird, enthaelt die entsprechenden Daten unveraendert - so wie sie in der Eingabedatei enthalten sind. Attribute wie NUMERIC-ASCII und DESCENDING veraendern die Ursprungsdaten.

Das interne 9-Byte-Format, erzeugt durch das NUMERIC-ASCII-Attribut, ist eine sortierbare BCD-Gleitkommazahl:

Bytes/Bits	Definitionen
Byte 0: Bit 7	1
Byte 0: Bit 6-0	normalisierter Exponent (Basis 10), berechnet aus: Wert der Ziffer nach E, wenn vorhanden, plus der Zahl der Ziffern vor dem Dezimalpunkt, plus 512; Null, wenn keine signifikanten Ziffern vorhanden sind
Bytes 2 - 6	signifikante Ziffern, jede Ziffer 4 Bit, nach rechts mit 0 aufgefuellt

War die Zahl negativ, wird das Einerkomplement gebildet.

Das FLOATING-POINT-Attribut transformiert den Schluessel wie folgt:

1. umgekehrte Bytereihenfolge
2. wenn Byte 0 (nach der Umkehrung) 0 ist, werden alle Bytes auf 0 gesetzt
3. Bit 0 von Byte 0 erhaelt Wert von Bit 7 Byte 1
4. Byte 0 um eine Position nach rechts verschoben
5. Bit 7 von Byte 0 wird 1 gesetzt
6. wenn Bit 7 von Byte 1 vor Schritt 3 1 war, Einerkomplement aller Bytes

4.2. R-OUTPUT (Satznummern)

Eine R-OUTPUT-Datei enthaelt nur die Eingabesatznummern. Die Satznummern beginnen bei 1, die maximale Satznummer ist 65535. Wird die Ausgabedatei als CR-DELIMITED spezifiziert, werden die Nummern als ASCII-Zeichen mit fuehrenden Nullen zum Auffuellen auf 6 Ziffern ausgegeben.
Ist die Ausgabedatei als FIXED-LENGTH, VARIABLE oder RELATIVE deklariert, werden die Satznummern binaer ausgegeben, je 3 Byte lang. Das erste Byte ist stets 0, das zweite Byte der L-Teil der Satznummer und das dritte Byte der H-Teil.
Wird eine FIXED-LENGTH R-OUTPUT-Datei mit einem anderen Programm gelesen, wird das Dateiende durch das physische Dateiende oder durch 1AH im ersten Byte (oder FFH beim FFZZZ-Ausgabeattribut) angezeigt.

4.3. P-OUTPUT (Zeiger)

Eine P-OUTPUT-Datei enthaelt "Zeiger" zum Beginn eines jeden Eingabesatzes. Jeder Zeiger besteht aus zwei Zahlen: der Sektornummer (beginnend bei 1), die den Sektor bezeichnet, in dem der Satz beginnt und einem Byte-Offset (Zahl zwischen 1 und 128), das anzeigt, mit welchem Byte im Sektor der Satz beginnt. Ist die Ausgabedatei CR-DELIMITED, werden die Zahlen als ASCII-Zeichen ausgegeben, 6 Ziffern fuer jeden Satz, durch Kommas getrennt.
Ist die Ausgabedatei von anderem Typ, so werden die Zahlen als 2-Byte-Werte binaer ausgegeben, der L-Teil zuerst.
Wird eine FIXED-LENGTH-P-OUTPUT-Datei mit einem Programm gelesen, kann das Dateiende entweder durch das physische Dateiende oder durch einen Satz, dessen beide Zahlen aus 1AH bestehen, erkannt werden (ist die Datei mit FFZZZ geschrieben, dann -1 als erste Zahl).

4.4. KR-OUTPUT (Schluessel und Satznummern)

Eine KR-OUTPUT-Datei enthaelt die Schluessel, gefolgt von den Satznummern, wie oben beschrieben.

4.5. KP-OUTPUT (Schluessel und Zeiger)

Eine KP-OUTPUT-Datei enthaelt die Schluessel, gefolgt von den Zeigern zu den Eingabesaetzen, wie oben beschrieben.

2.11. ANHANG B: Fehlerausschriften

Kommandoeingabe

Bei einem Fehler waehrend der Kommandoeingabe wird die erkannte fehlerhafte Position durch ein Zeichen ^(5EH) gekennzeichnet, eine Fehlermeldung ausgeschrieben und die Bereitschaft zur erneuten Kommandoeingabe angezeigt.
Tritt ein Fehler in einer Zeile mit mehreren Kommandos auf, so sind die rechts vom fehlerhaften Kommando stehenden Kommandos zu wiederholen, die links stehenden bleiben gueltig.

GO

Bei Fehlerausschriften nach dem GO-Kommando erfolgt die Ruekkehr zur Kommandoeingabe, wo fehlende Kommandos ergaenzt bzw. fehlerhaft eingegebene Kommandos korrigiert werden koennen. Vorher eingegebene korrekte Kommandos bleiben gueltig.

Sortier-/Mischlauf

Fehler waehrend der Programmausfuehrung fuehren zum Abbruch. Die benutzten Dateien werden nicht geschlossen, die Arbeitsdatei name.□□□ wird nicht geloescht.
Normalerweise wird nach einem solchen Fehler zum Betriebssystem zurueckgekehrt. Wurde das RETURN-TO-CONSOLE-Kommando gegeben, wird mit einer Kommandoeingabe weitergearbeitet. Vorher eingegebene Kommandos sind ungueltig.

Im folgenden werden die vollstaendigen Fehlerausschriften angegeben; die Fehlernummer ist Bestandteil der Ausschrift.
Wurde das NO-ERROR-MESSAGES-Kommando eingegeben, werden nur "ERROR S" und die Fehlernummer ausgeschrieben.

ERROR S29: NOT ENOUGH WORKING STORAGE

erscheint nach GO; die Groesse des verfuegbaren Arbeitsspeichers reicht fuer die spezielle Kombination von Satzlaengen und Optionen nicht aus

**ERROR S30: COMMAND INPUT WORKSPACE FULL,
TRY MORE MEMORY OR SIMPLER COMMANDS**

tritt waehrend der Kommandoeingabe auf; das eingegebene Kommando ist zu komplex fuer den verfuegbaren Speicherbereich, um ausgefuehrt werden zu koennen

ERROR S37: INVALID FILE NAME

fehlerhafter Dateiname waehrend der Kommandoeingabe: enthaelt keine Buchstaben oder Ziffern vor der Typangabe (wenn vorhanden) bzw. die Zeichen "*" oder "?"

ERROR S38: INVALID DRIVE NAME

unkorrekte Angabe des Laufwerks in einem Kommando

ERROR S39: UNRECOGNIZED COMMAND

Das erste Wort des Kommandos ist kein Schluesselwort bzw. keine zulaessige Abkuerzung eines Schluesselwortes.

ERROR S40: INVALID ARGUMENT

falsches Argument bei Kommandoeingabe

ERROR S41: MISSING ARGUMENT

fehlendes Argument bei Kommandoeingabe

ERROR S42: INVALID LINE TERMINATION

Kommandozeile fehlerhaft abgeschlossen

ERROR S43: NUMBER REQUIRED

fehlende Ziffer bei erforderlicher Zahlenangabe im Kommando

ERROR S44: NUMBER TOO LARGE

Zahl im Kommando groesser als 65535 (ausser numerische Konstanten in SELECT- und EXCLUDE-Argumenten)

ERROR S45: RECORD LENGTH MISSING

Kommandoeingabe: fehlende Satzlaenge im INPUT-ATTRIBUTES-Kommando

ERROR S46: THAT COMMAND MUST BE LAST ON LINE

Kommandoeingabe: "GO" und "CFILE" duerfen kein ";" und ein weiteres Kommando folgen

ERROR S50: NOT INPUT FILE SPECIFIED

nach GO-Kommando: kein SORT-FILES- oder MERGE-FILES-Kommando angegeben; nach der Spezifikationen der fehlenden Eingabedatei kann erneut das GO-Kommando eingegeben werden

ERROR S51: MORE THAN 32 SORT INPUT FILES

mehr als 32 Eingabedateien

ERROR S52: NO KEYS SPECIFIED

nach GO-Kommando: kein gueltiges KEY-Kommando eingegeben

ERROR S53: MORE THAN 32 KEYS

mehr als 32 Schluessel vorgegeben

ERROR S54: MORE THAN 32 SELECT CRITERIA

Kommandoeingabe: Anzahl der SELECT-Kommandos plus Anzahl der EXCLUDE-Kommandos ist groesser als 32

ERROR S55: NO OUTPUT FILE SPECIFIED

nach GO-Kommando: kein OUTPUT-FILE-Kommando eingegeben

ERROR S56: PRINT LEVEL NOT 0 TO 5

waehrend Kommandoeingabe; keine exakte Vorgabe der Ausschriftsvariante

ERROR S57: RECORD LENGTH NOT 1 TO 4096

waehrend Kommandoeingabe: falsche Ein- oder Ausgabedateilaenge; bei VARIABLE-Dateien ist die minimale Satzlaenge 3

ERROR S58: FIELD END COLUMN < START COLUMN

Kommandoeingabe: bei Positionsfeldspezifikationen in KEY-, SELECT- und EXCLUDE-Kommandos, Feldende < Feldanfang

ERROR S59: #-FIELD NUMBER NOT 1 TO 255

Kommandoeingabe: bei KEY-, SELECT- und EXCLUDE-Kommafeldern, Feldnummer nicht im Bereich 1 bis 255

ERROR S61: FIELD END COLUMN > RECORD LENGTH

Kommandoeingabe oder nach GO: Positionsfeldangabe bei KEY, SELECT und EXCLUDE, Feldende > Satzlaenge

ERROR S62: FIELD START COLUMN < 1 OR > RECORD LENGTH

Kommandoeingabe oder GO: Positionsfelder bei KEY, SELECT und EXCLUDE, Feldanfang < 1 oder > Satzlaenge

ERROR S63: FIELD LENGTH NOT 1 TO 4096

Eingabe eines KEY-, SELECT- oder EXCLUDE-Kommandos, Feldlaenge nicht im Bereich 1 bis 4096

ERROR S65: NO 'GO' AFTER ERROR IN COMMAND
FROM COMMAND FILE —
CORRECT ERROR, ENTER 'GO' AGAIN

erscheint nach Eingabe eines GO-Kommandos, wenn in einem vorher eingegebenem Kommando ein Fehler enthalten ist (z.B. bei einer CFILE-Kommandodatei)

ERROR S67: TAGSORT WITH INPUT FILE SAME AS OUTPUT

GO-Kommando: TAGSORT angegeben und als Ein- und Ausgabedatei gleiche Datei spezifiziert

ERROR S68: MORE THAN 32 MERGE-ONLY INPUT FILES

waehrend Kommandoeingabe: mehr als 32 Mischeingabedateien

ERROR S69: NO INPUT RECORD LENGTH SPECIFIED

nach GO-Kommando: kein INPUT-ATTRIBUTES-Kommando eingegeben

ERROR S70:) MISSING

bei Eingabe eines SELECT- oder EXCLUDE-Kommandos: Klammern fehlen dort, wo sie entsprechend der Syntax erforderlich sind

ERROR S71: COMPARISON OPERATOR REQUIRED

bei Eingabe eines SELECT- oder EXCLUDE-Kommandos: nach der Angabe eines Wertes (Feldspezifikation, Textkonstante oder numerische Konstante) fehlt ein Vergleichsoperator

ERROR S72: FIELD OR CONSTANT REQUIRED

bei Eingabe eines SELECT- oder EXCLUDE-Kommandos: vor oder nach Vergleichsoperator fehlt Wertangabe (Feld, Text, Zahl)

ERROR S73: SECOND " MISSING

Kommandoeingabe: ein abschliessendes Anfuehrungszeichen (") fehlt bei einer Textkonstanten

ERROR S74: ILLEGAL DIGIT FOR NUMBER BASE

bei Eingabe eines SELECT- oder EXCLUDE-Kommandos: Zeichen 8 und 9 oder A bis F sind in einer Oktalkonstanten bzw. A bis F in einer Dezimalkonstanten enthalten

ERROR S75: SIZE IN BYTES IS TOO SMALL TO HOLD THE VALUE GIVEN

bei Eingabe eines SELECT- oder EXCLUDE-Kommandos: die signifikanten Ziffern einer Konstanten erfordern zur Speicherung mehr Bytes als nach der Basisangabe spezifiziert sind, z.B. sei 12345H2 spezifiziert, so erfolgt eine Fehlerausschrift, da 12345 hexadezimal nicht in 2 Bytes gespeichert werden kann

ERROR S76:] MISSING "

bei Eingabe eines SELECT- oder EXCLUDE-Kommandos: weder eine andere gueltige Konstante noch die schliessende Klammer (]) in einer in eckigen Klammern [ ] einzuschliessenden Konstantenliste gefunden

ERROR S78: 1-BYTE CONSTANT REQUIRED

bei Eingabe eines COLLATING-SEQUENCE-Kommandos: als Zeichen oder Positionswert werden zwei oder mehr in Anfuehrungszeichen eingeschlossene Zeichen (bzw. nur "") angegeben oder als Zahl, deren Wert groesser als 255 ist

ERROR S79: TOO MANY NESTED ()'s OR INTERNAL ERROR

bei Eingabe eines SELECT- oder EXCLUDE-Kommandos: mehr als 5 verschachtelte Klammern oder interner Fehler

**ERROR S81: WORKING STORAGE FULL:
TRY USING MORE WORKING STORAGE, OR TRY TAGSORT.**

waehrend Programmausfuehrung: eine relativ grosse Anzahl zu sortierender oder zu mischender Eingabesaetze bei einem kleinen verfuegbaren Arbeitsspeicher fuehrt zum Ueberlauf des Arbeitsspeichers

ERROR S85: MERGE-ONLY INPUT FILE NAME SAME AS OUTPUT FILE

nach GO-Kommando: Ausgabedatei bei Mischen darf keine der Eingabedateien sein

ERROR S86: TAGSORT, AND INPUT FILE DRIVE SAME AS OUTPUT /C DRIVE

nach GO-Kommando, wenn bei TAGSORT und Wechsel der Ausgabediskette fuer Ein- und Ausgabedatei das gleiche Laufwerk angegeben wurde

ERROR S87: WORK FILE DRIVE SAME AS OUTPUT /C DRIVE

bei GO-Kommando: wenn Wechsel der Ausgabediskette spezifiziert wurde, muss die Arbeitsdatei einem anderen Laufwerk zugeordnet sein (die Arbeitsdatei ist dem aktuellen Laufwerk zugeordnet, wenn nicht mit dem WORK-DRIVE-Kommando anderweitig spezifiziert)

ERROR S88: INVALID COMBINATION OF FIELD TEST ATTRIBUTES

eine unzulaessige Kombination von KEY- oder SELECT-/EXCLUDE-Feldtestattributen wurde spezifiziert, so z.B. NUMERIC-ASCII und INTEGER zusammen oder UPPER-CASE und FLOATING-POINT; auch wenn LOHI fuer ein Kommafeld angegeben wurde;
Fehlerausschrift erscheint nach Kommandoeingabe

ERROR S89: MORE THAN ONE OF CR-DELIMITED, FIXED, VARIABLE, AND RELATIVE

zwei verschiedene Dateitypen fuer Eingabe (oder Ausgabe) spezifiziert (Eingabe und Ausgabe koennen aber verschiedenen Typs sein)

ERROR S91: MORE THAN ONE INPUT FILE WITH OUTPUT OPTION OR TAGSORT

R-OUTPUT, P-OUTPUT, KR-OUTPUT und KP-OUTPUT duerfen nur mit einer (Sortier- oder Misch-) Eingabedatei verwendet werden; TAGSORT darf nur mit einer Sortiereingabedatei benutzt werden. (Bei K-OUTPUT koennen mehrere Eingabedateien angegeben werden.)

ERROR S92: MORE THAN ONE OUTPUT OPTION, OR AN OUTPUT OPTION AND TAGSORT

nach GO: mehr als eine der Ausgabemoeglichkeiten K-, R-, P-, KP-, KR-OUTPUT oder TAGSORT spezifiziert

ERROR S94: MERGE-ONLY INPUT FILE WITH TAGSORT

nach GO: TAGSORT darf nur bei Angabe einer Sortiereingabedatei verwendet werden

ERROR S128: FILE name.typ NOT FOUND

bei SORT-FILES- oder MERGE-FILES-Kommando: angegebene Datei im spezifizierten Laufwerk nicht gefunden

ERROR S129: DISK d: FULL WHILE WRITING FILE name.typ

Erscheint weahrend der Programmausfuehrung, wenn fuer die gesamte Datei nicht genuegend freier Speicherplatz vorhanden ist. Ist diese Datei die Arbeitsdatei (name.typ als SORT.▫▫▫ angezeigt), kann mit dem WORK-DRIVE-Kommando ein anderes Laufwerk zugewiesen bzw. TAGSORT zur Reduzierung der Groesse der Arbeitsdatei benutzt werden.

ERROR S130: DIRECTORY OF DISK d: FULL WHILE WRITING FILE name.typ

waehrend Sortierlauf: Diskettenverzeichnis gefuellt

ERROR S131: CLOSE FAILURE FILE name.typ

interner Fehler: Diskette waehrend des Laufes gewechselt

ERROR S133: OVERLONG RECORD (LENGTH nnn) IN FILE name.typ

erkannte Satzlaenge in einer VARIABLE-Eingabedatei laenger als die spezifizierte maximale Satzlaenge

ERROR S134: NOT A RELATIVE FILE : FILE name.typ

Eingabedatei als RELATIVE spezifiziert, aber kein Dateikennsatz (mit einer Null im ersten Byte) vorhanden

ERROR S135: INCORRECT RECORD LENGTH GIVEN FOR RELATIVE FILE name.typ

waehrend Programmausfuehrung: spezifizierte Eingabesatzlaenge entspricht nicht der im Dateikennsatz angegebenen

ERROR S140: COMMAND DECODING WORKSPACE FULL, TRY MORE MEMORY OR SIMPLER COMMANDS

waehrend Kommandoeingabe: entweder ein sehr kompliziertes Kommando (wahrscheinlich bei SELECT oder EXCLUDE) wurde bei kleinem Arbeitsspeicherbereich eingegeben oder ein interner Fehler

ERROR S160: INTERNAL ERROR

ERROR S253: INTERNAL ERROR

interner Fehler

ERROR Snn: NO SUCH ERROR

ERROR-MESSAGE-Kommando mit undefinierter Fehlernummer eingegeben oder interner Fehler.

3. Universelles Dienstprogramm DIENST 1520 (SCPX)

3.1. Einleitung

DIENST ist ein kompaktes Programmpaket der im System enthaltenen Dienst- und Manipulationsfunktionen. Neben auch in anderen Dienst- und Hilfsprogrammen enthaltenen Funktionen werden eine Reihe zusaetzlicher Moeglichkeiten (z.B. Lesen/Schreiben ab Spur/Sektor, Diskettentest, Wiederzugaenglichmachung von geloeschten Dateien) realisiert bzw. ist durch das Nummernmenue ein effektiveres Arbeiten mit bereits vorhandenen Funktionen moeglich. Das Programm belegt 16K.

Kommandouebersicht:

CHECK	Kontrollsummenberechnung von Dateien
CM	Byteweiser Vergleich von Hauptspeicherbereichen
COPY	Kopieren von Dateien
DIR	Auflisten der Dateibezeichnungen des Verzeichnisses (directory) einer Diskette auf dem Bildschirm
DISK	Anzeige von Diskettensystemparametern
DS	Byteweise Anzeige und Aenderung von Hauptspeicherinhalten
DUMP/DUMPX/ DUMPH/DUMPA	Anzeige von Hauptspeicherinhalten (in verschiedenen Codes)
ERA	Loeschen von Dateien
EXIT	Verlassen des DIENST-Programmes und Rueckkehr in den SCPX-Grundzustand
FILL	Speicherbereich mit vorgegeben Zeichen auffuellen
GO	Laden und Starten eines Programmes (mit Adressangabe)
GROUP	Auflisten der durchnumerierten 2K-Bloecke der Diskettendateien
JP/EX	Starten von Programmen
LOAD	Laden von Dateien
SAVE	Speicherinhalt als Datei auf Diskette aufzeichnen
LOG	Anzeige der aktuellen DIENST-Standardparametereinstellungen

MOVE	Verschiebung von zusammenhaengenden Speicherbereichen
READ	Lesen von Diskette (ab Spur/Sektor) in den Speicher oder auf den Bildschirm
WRITE	Aufzeichnen von im Speicher stehenden Daten auf Diskette (ab Spur/Sektor)
READGR	Lesen von Diskette (ab Blocknummer) in den Speicher oder auf den Bildschirm
WRITEGR	Aufzeichnen von im Speicher stehenden Daten auf Diskette (ab Blocknummer)
RECLAIM	Wiederzugaenglichmachung von aus dem Verzeichnis geloeschten Dateien
REN	Umbenennung von Dateien
RESET	Ruecksetzen einzelner Laufwerke in den Grundzustand
RUN	Laden und Starten von Programmen
SEARCH	Suchen von definierter Bytefolge im angegebenen Hauptspeicherbereich
SET	Setzen von Merkmalen in Dateibezeichnung
SETDIR/ SETSYS	Wandlung DIR-Dateien (--) SYS-Dateien
SET RO/ SETRO	Wandlung schreibgeschuetzte (--) nichtschreibgeschuetzte Dateien
SORT	Sortieren der Verzeichniseintragungen
SIZE	Meldung ueber die Groesse von Dateien
SPEED	Festlegung der Bildschirmausgabegeschwindigkeit
STAT	Kapazitaetsmeldung ueber freien bzw. belegten Speicherbereich einer Diskette
TEST	Test von Disketten
TYPE/TYPEX/ TYPEH/TYPEA	Auflisten von Dateien auf den Bildschirm (in verschiedenen Codes)
USER	Einstellen Benutzerbereich
XUSER	Festlegung Zielbenutzerbereich beim Kopieren

UR Anspringen spezieller zusaetzlich vom Anwender in DIENST eingebundener Routinen

Der Aufruf von DIENST erfolgt durch Eingabe von:

A>DIENST

Das Programm wird geladen, und es erscheint die Ausschrift:

DIENST 1520 (SCPX) V 1.0

WESENTLICHE SCP-DIENSTFUNKTIONEN

A0=

A0 = ist die Bereitschaftsmeldung zur Kommandoeingabe (A ist das Laufwerk, von dem DIENST aufgerufen wurde, 0 ist das User-Feld). Wird ein anderes Laufwerk (LW) gewuenscht, kann dies mit der Eingabe der LW-Bezeichnung (A-E) angegeben werden (z.B.: A0=b:, es erscheint B0=. B ist somit das aktuelle Laufwerk).

Das Kommando zur Rueckkehr in das SCP-Kommandoeingabeniveau ist **EXIT.**

Werden waehrend der Arbeit mit DIENST in einem Laufwerk die Disketten gewechselt, dann ist ein Warmstart auszufuehren, um die Diskette R/W zu setzen. Andernfalls erscheint bei Schreibzugriff eine entsprechende Aufforderung auf dem Bildschirm.

Zum Abbruch eines Kommandos in DIENST wird ^C oder die **ESCAPE**-Taste verwendet.

Alle DIENST-Kommandos koennen durch Eingabe eines Fragezeichens (?) angezeigt werden.

z.B.: (A0=?)

Will man nur ein DIENST-Kommando abarbeiten und dann in das SCP-Kommandoeingabeniveau zurueckkehren, gibt man ein:

A>DIENST Kommando
z.B.: A>DIENST CHECK

Bei einigen Kommandos koennen zusaetzliche Parameter, in eckigen Klammern eingeschlossen, angegeben werden.

z.B.: A0=COPY **[Q]** Q - Umbenennen des Dateinamens (hier waehrend des Kopierens)

Fehlen die Parameter, gelten die mit dem DIENST-Kommando **LOG** anzeigbaren Standardeinstellungen.

Achtung: In Laufwerk A muss immer eine Systemdiskette liegen!

3.2. Kommandoeingabesyntax des DIENST

3.2.1. Das DIENST-Nummernmenue

Einige Kommandos zeigen bei ihrem Aufruf das Dateiverzeichnis oder Teile daraus (Dateigruppen) an, wobei die Dateien mit einer laufenden Nummer versehen sind. Diese Nummer kennzeichnet die Datei fuer die weitere Arbeit mit Kommandos.
Nach der Frage "**WAEHLE?**" koennen eine, mehrere oder Gruppen von Dateien angegeben werden.
Einzelne Nummern sind durch Leerzeichen zu trennen; Bereiche werden durch Anfang und Ende mit Minus (-) verbunden angegeben. Fehlt die Angabe des Endes, wird bis zur letzten Nummer gewaehlt.

Beispiel: **WAEHLE?** 1 3 4-8 10-

Die zugewiesenen Nummern haben keine bleibende Verbindung zu den Dateien, sie koennen veraendert werden (z.B: durch Loeschen oder Umsortieren von Dateien).

Bei der Eingabe der Kommandos, die das Nummernmenue aufrufen, gibt es folgende Moeglichkeiten (als Beispiel dient REN):

(1) A0=REN <ENTER> - Anzeige Nummernmenue aktuelles LW A
(2) A0=REN B: <ENTER> - Anzeige Nummernmenue von LW B
aktuelles LW bleibt A
(3) A0=B:REN <ENTER> - Umschalten aktuelles LW von A auf B
Anzeige Nummernmenue von LW B
(4) A0=B:REN C: <ENTER> - Umschalten aktuelles LW von A auf B
Anzeige Nummernmenue von LW C
(5) A0=REN [ ] <ENTER> - Anzeige Nummernmenue laut Parametern
(z.B: Spaltenangabe)--> siehe LOG-Kommando!
(6) A0=REN T*.* <ENTER> - Anzeige Nummernmenue der Dateien,
deren Namen mit T beginnt
(*.* - allgemeine SCP-Dateibezeichnung)
(7) A0=REN T** <ENTER> - wie (6), aber die DIENST-Dateibezeichnung nutzend
(8) A0=REN *** <ENTER> - Anzeige Nummernmenue aller Dateien
des LW A
(*** -DIENST-Bezeichnung fuer beliebige Dateien)
(9) A0=REN S?Z?.COM
<ENTER> - Anzeige aller .COM-Dateien, deren
Namen 4 Zeichen lang sind und an
erster Stelle ein S und an dritter
Stelle ein Z haben

3.2.2. Kommandosyntax ohne Nummernmenue (mit Dateinamen)

Bei dieser Kommandoeingabe wird die exakte Dateibezeichnung verwendet.

Es gibt folgende Moeglichkeiten (als Beispiel dient TYPE):

(1) A0=TYPE XXX.DOK <ENTER> - Kommando zur Ausgabe der Datei XXX.DOK von LW A

(2) A0=TYPE B:XXX.DOK <ENTER> - wie (1), aber von LW B, A bleibt aktuelles LW

(3) A0=B:TYPE XXX.DOK <ENTER> - Umschalten aktuelles LW von A auf B, Ausgabe der Datei von B

3.2.3. Kommandosyntax ohne Nummernmenue und ohne Dateinamen

Bei einigen Kommandos ist nur die genaue Kommandobezeichnung anzugeben. Es gibt folgende Moeglichkeiten (als Beispiel dient TEST):

(1) A0=TEST <ENTER> - Test Diskette in LW A

(2) A0=TEST B: <ENTER> - Test Diskette in LW B, aktuelles LW bleibt A

(3) A0=B:TEST <ENTER> - Umschaltung aktuelles LW von A auf B, Test Diskette in LW B

3.3. Steuertasten

3.3.1. Steuertasten fuer Bildschirm

Leertaste oder ^S:

- sofortiges Beenden aller Ausgabeoperationen
- es kann folgen:
  Leertaste noch einmal --> Zeilenvorschub
  <ENTER> --> Anzeige neue Bildschirmseite
  andere Taste --> fortlaufende Anzeige

ENTER oder ^M, ^J:

- zum Vorschub (z.B. bei Auflisten eines grossen Diskettenverzeichnisses)
- es wird jeweils eine neue Bildschirmseite ausgegeben

^P:

- Druckereinschaltung (nur fortlaufend, nicht seitenweise moeglich)
- eine im CCP-Grundzustand erfolgte Druckerzuschaltung kann innerhalb DIENST nicht rueckgaengig gemacht werden
  (Achtung: Bei nochmaligem ^P erfolgt Zeichendruck doppelt!)

0-9:

- Eingabe einer dieser Nummern waehrend der Ausgabe macht seitenweise Ausgabe ungueltig und setzt Geschwindigkeit der zeilenweisen Ausgabe (0-schnell, 9-langsam)
- siehe SPEED-Kommando

Beliebige andere Taste:

- macht seitenweise Ausgabe ungueltig
- zeilenweise fortlaufende Ausgabe

3.3.2. Steuertasten fuer Kommandoeingabe

DIENST besitzt einen Mini-Editor, der es erlaubt, Eingabefehle zu loeschen und zu korrigieren. Ausserdem arbeiten die herkoemmlichen SCP-Steuerzeichen auch im DIENST-Programm.

DELETE/BACKSPACE/^H	- Loeschen letztes eingegebenes Zeichen, danach Neueingabe moeglich
ENTER ^M/^J	- Abschluss Kommandoeingabe und Ausfuehrungsbeginn
^E	- Kursor auf neue Zeile setzen (bei langen Kommandoanweisungen)
^P	- Druckerzuschaltung
^R	- nochmalige Anzeige einer eingegebenen und korrigierten Eingabezeile
^U	- Ungueltigmachen der Kommandoeingabe, Kursor auf Anfang neue Zeile setzen
^X	- Ungueltigmachen der Kommandoeingabe, Kursor auf Anfang gleiche Zeile setzen

3.4. Kommandos des DIENST

3.4.1. ? (Hilfsfunktion)

A0=?

Es wird eine Liste aller DIENST-Kommandos auf dem Bildschirm angezeigt. Als Hinweis auf dieses Kommando erscheint bei fehlerhaften Kommandoeingaben (Syntaxfehler) die Bildschirmausschrift "incorrect, for list of commands enter (?)".

3.4.2. CHECK

A0=**CHECK**

CHECK dient zum Errechnen von Kontrollssummen einzelner Dateien. Der Anwender kann schnell pruefen, ob zwei Dateien identisch sind. Fuer weitere Untersuchungen im Falle Nichtidentitaet kann das Speichervergleichskommando CM zum Finden unterschiedlicher Bytes genutzt werden, oder man listet mit dem TYPE-Kommando die Dateien komplett auf Bildschirm und Drucker aus. Die laufende Summe der Dateikontrollsummen kann fuer den Vergleich von Dateigruppen auf Identitaet herangezogen werden. Fuer die Dateiauswahl kann die Syntax des Nummernmenues verwendet werden. Die berechneten Kontrollsummen zur jeweiligen Datei werden auf dem Bildschirm angezeigt.

Beispiel:

```
A0=CHECK (ENTER)
A0: 1= DOK      .DOK |  2= TEST     .COM |  3= TKO      .DOC
A0: 4= DIENST   .COM |  5= DIENST   .DOK
WAEHLE? 2 3

A: TEST     .COM -checksum: D1D8 total: D1D8 dec: 53723
A: TKO      .DOK -checksum: 368B total: 0866 dec:  2150
A0=
```

checksum - aktuelle Kontrollsumme der Diskettendatei in HEX-Darstellung
total - laufende Summe der Kontrollsummen in HEX
dec - total-Wert in DEZ

3.4.3. CM

A0=**CM (aadr1) (eadr1) (aadr2)**

(aadr1) - Anfangsadresse erster Speicherbereich in HEX
(eadr1) - Endadresse erster Speicherbereich in HEX
(aadr2) - Anfangsadresse zweiter Speicherbereich in HEX

Das Kommando dient zum byteweisen Vergleich zweier Speicherbereiche. Die Laenge der zu vergleichenden Bereiche wird automatisch aus der Differenz von End- und Anfangsadresse des ersten Speicherbereiches berechnet. Deshalb erfolgt keine Eingabe der Endadresse des zweiten Speicherbereiches. Das Kommando ermoeglicht die Kontrolle zweier Dateien auf Identitaet, wenn vorher mit dem DIENST-Kommando LOAD diese Dateien auf die entsprechenden Bereiche gebracht wurden. Es werden nur die unterschiedlichen Bytes mit Adressen und Inhalt angezeigt.

Beispiel:

```
A0=CM 4000 4003 5000 <ENTER>
4000  43  54  5000
4002  44  55  5002
(Adr. Inh.Inh. Adr.)
```

3.4.4. COPY

A0=**COPY [par]**

COPY dient zum Kopieren einer beliebigen Anzahl von Dateien auf ein anderes Laufwerk oder auf die Quelldiskette unter anderem Namen (Parameter Q angeben). In Verbindung mit dem Kommando XUSER koennen auf der Quelldiskette in anderen Nutzerbereichen Kopien abgelegt werden (siehe 3.4.37.). COPY kann mit dem Nummernmenue arbeiten, so dass nur die Nummern der zu kopierenden Dateien und das Ziellaufwerk eingeben werden muessen. Zur Kontrolle wird der Name der im Kopiervorgang befindlichen Datei auf dem Bildschirm angezeigt. Erscheint eine Datei, die nicht kopiert werden soll, so drueckt man die ESCAPE-Taste, wodurch der Kopiervorgang abgebrochen wird. Auf der Zieldiskette befindet sich dann eine Pseudodatei "Name.¤¤¤", die geloescht werden muss. Bei diesem Kommando ist die Angabe von Parametern in [ ] moeglich. Vor dieser Eingabe ist deshalb das LOG-Kommando zu aktivieren, um zu kontrollieren, wie der Standard gesetzt ist. Vor dem Kopieren kann mit SIZE und STAT herausgefunden werden, ob die zu kopierenden Dateien noch auf der Zieldiskette Platz haben. Wird beim Kopieren BAD SECTOR festgestellt, so erscheint die Abfrage "bad sector on read, abort (Y/N):"(Abbruch des Kopiervorgang JA/NEIN). Bei Eingabe N wird das Kopieren fortgesetzt, wobei der Anwender anschliessend mit TYPE die kopierte Datei auf der Zieldiskette kontrollieren sollte.

Wichtige COPY-Parameter [par]:

A - Existiert die Datei auf dem Ziellaufwerk bereits, dann Ueberschreiben dieser (**overlay**)

B - Existiert die Datei auf dem Ziellaufwerk bereits, dann automatisches Wandeln in eine BAK-Datei und Kopieren der neuen Datei (**back up**)

C - Ist die Datei auf dem Ziellaufwerk bereits vorhanden, dann Frage ob: (**O**) verlay, (**B**) ack up or (**S**)kip:
(Eingabe O, B oder S --> siehe A-, B- und D-Parameter)

D - Existiert die Datei auf dem Ziellaufwerk bereits, dann kein Kopieren (**skip**)

R (ON) - (Y/N)-Entscheidungsfrage vor Kopieren jeder einzelnen Datei

V (ON) - Lesen der kopierten Datei zur Kontrolle auf Schreibfehler

T (ON) - Abbruch des COPY beim Kopieren einer Dateiserie, bei jener Datei in der eingegebenen Reihenfolge, die nicht auf die Zieldiskette passt

T (OFF) - Passt die Dateiserie nicht komplett auf die Zieldiskette, werden soviel Dateien wie moeglich uebernommen (d.h. der zu grossen Datei folgende kleinere Dateien, die noch Platz haben, werden kopiert)

M (ON) - Markieren der Dateien im Verzeichnis nach dem Kopieren --> < Originaldatei (auf Quelldiskette) (Marke kann durch Kommando **SET [+2]** geloescht werden)

--> > kopierte Datei (auf Zieldiskette) (Marke kann durch Kommando **SET [+1]** geloescht werden)

Q - erlaubt Umbenennen der Dateien, die kopiert werden damit z.B. sofortiges Anlegen eines Duplikates einer Datei auf der gleichen Diskette moeglich ist

Die Parameter mit (ON/OFF) schalten ihre Wirkung um, wenn sie ein zweites Mal angegeben werden.
Es koennen mehrere Parameter gleichzeitig angegeben werden (z.B: A0=**COPY [CROM]**).

Die Parameter A, B, C und D schliessen einander aus.

3.4.5. DIR

A0=**DIR [UXI]**

Das Kommando dient zur Anzeige von Bibliotheksverzeichnissen. Es besteht die Moeglichkeit, die Verzeichnisse aller Laufwerke und USER-Felder aufzulisten.

U - alle USER-Felder (wenn nicht angegeben, dann nur aktuelles USER-Feld)
X - alle Laufwerke (wenn nicht angegeben, dann nur aktuelles Laufwerk)
I - Anzahl der Anzeigespalten (wenn nicht angegeben, dann entsprechend Spaltenangabe bei LOG)

Beispiel: A0=DIR [UX2]

Die Parameter koennen auch einzeln oder gar nicht angegeben werden.
Durch Eingabe mehrdeutiger Dateibezeichnungen koennen auch bestimmte Dateigruppen angezeigt werden. Dabei wird die SCP-Bezeichnung (*.*) ebenso akzeptiert, wie die spezielle DIENST-Bezeichnung (**).

- gleicher Dateityp: **DIR Name**** (SCP: DIR Name.*)
- gleicher Dateiname: **DIR **Typ** (SCP: DIR *.Typ)

Die angezeigten Dateien koennen folgende Kennzeichnung besitzen:

* - Datei ist R/O gesetzt
( ) - Systemdatei
(* - R/O und Systemdatei

3.4.6. DISK

A0=**DISK**

Dieses Kommando zeigt die wichtigsten Systemparameter der Disketten (Diskettenformat, Kapazitaet, Dichte, Informationen ueber Systemspur) an.

Beispiel:

```
Disk:              A:                    ;Disketten-LW A
Disk capacity:     296K                  ;Diskettenkapazitaet
Tracks:             77    3 System       ;77 Spuren,
                                          davon 3 Systemspuren
Sectors/tracks:     32   32 last         ;32 Sektoren/Spur, letzte
                                          Spur hat 32 Sektoren
Sectors/system:     96   16 dir          ;96 Sektoren fuer System
                                          und 16 Sektoren fuer
                                          Verzeichnis
Dir entries:        64    2K            ;Verzeichnis kann 64
                                          Eintragungen aufnehmen
                                          (auf 2K)
Sectors/group:      16    2K 93H group    ;Block aus 16 Sektoren be-
                                          legt 2K; groesstmoegli-
                                          che Blocknummer (in HEX)
                                          auf der Diskette = 93H
KByte/extent:       32K                   ;ein Extent belegt 32K
                                          (gemeint ist die maximale
                                          Dateigroesse, auf die
                                          sich eine Verzeichnisein-
                                          tragung bezieht)
```

3.4.7. DS

A0=**DS adr**

Das Kommando dient zur byteweisen Anzeige und zur Aenderung von Speicherinhalten. Die Anzeige erfolgt ab der vorgegebenen Adresse in HEX-, DEZIMAL-, BINARY- und ASCII-Code. In der Spalte Enter wird die Codeart angezeigt, in dem das angezeigte Byte bzw. die folgenden Bytes geaendert werden koennen ((H), (D), (B) oder (A)). Dieser Eingabemodus kann durch nachfolgende Eingabe der gewuenschten Codeart umdefiniert werden (.H, .D, .B oder .A). Es koennen bis zu 120 Bytes, die

geaendert werden sollen, in gewuenschter Eingabemoduskombination aufeinanderfolgend und durch Leerzeichen getrennt eingegeben werden. Die Byteanzeige kann adresseitig vor- und rueckwaertsbewegend erfolgen (Umschaltung mit TILDE-Taste).

Beispiel:

```
A0=DS 5000 <ENTER>
addr: hex dec  binary    ascii  Enter
5000:  C3 195 11000011    -C     <A> <ENTER>
5001:  10  16 00010000    ^P     <A>.H 13 63 <ENTER>
5003:  1A  26 00011010    ^Z     <H> <~>
5002-  63  99 01100011     c     <H> <ENTER>
5001-  13  19 00010011    ^S     <H> <~>
5002:  63  99 01100011     c     <H> <ENTER>
5003:  1A  26 00011010    ^Z     <H>.A a b c .D 10 11 <ENTER>
5008:  2E  46 00101110     .     <D>.. <ENTER>
A0=
```

Bei ASCII-Code bedeutet das Minuszeichen (bei Adresse 5000: -C), dass das adresshoechste Bit gesetzt ist, das ^-Zeichen (z.B. ^P) verweist auf den Steuerfunktionscharakter eines Codes.

Erlaeuterung der Eingabezeichen

.A - Eingabemodus ASCII-Code
- -) es werden nur ASCII-Zeichen akzeptiert
- -) Zeichen sind durch ein Leerzeichen zu trennen
- -) Leerzeichen kann nicht durch ein ASCII-Zeichen angegeben werden (.H20 oder .D32 verwenden)
- -) eingegebene Steuerzeichen werden entweder ausgefuehrt (z.B. ^P) oder angezeigt (z.B. ^S)

.B - Eingabemodus Binaercode
- -) es werden nur Binaerwoerter akzeptiert
- -) Binaerwoerter sind durch Leerzeichen zu trennen
- -) alle 8-Bitpositionen muessen angegeben werden

.D - Eingabemodus Dezimalcode
- -) es werden nur Dezimalzahlen akzeptiert
- -) Dezimalzahlen sind durch Leerzeichen zu trennen

.H - Eingabemodus HEX-Code
- -) es wird nur der HEX-Code akzeptiert
- -) Hex-Zahlen sind durch Leerzeichen zu trennen

<ENTER> - Vorruecken auf neue Adresszeile
- -) wird nur der Eingabemodus geaendert, dann Anzeige der gleichen Adresszeile mit neuem Modus
- -) zu beachten: ^M und ^J erzielen die gleiche Wirkung wie <ENTER>

(~) - Umkehr der Richtung der Speicheranzeige
(TILDE) -) wenn Speicher vorwaerts durchschritten wird, erscheint Doppelpunkt nach der Adressanzeige
-) wenn Speicher rueckwaerts durchschritten wird, erscheint Minus (-) nach der Adressanzeige

..<ENTER> - beendet DS-Kommando

Ausserdem kann die Funktion genutzt werden, um von einem Zeichen alle Codedarstellungsarten anzuzeigen (z.B. den unbekannten Binaercode eines bestimmten ASCII-Zeichens), indem man das Zeichen auf einen unbelegten Speicherplatz eingibt und mit TILDE zu diesem Speicherplatz zurueckkehrt.

3.4.8. DUMP/DUMPX/DUMPH/DUMPA

Diese Kommandos dienen zur Inhaltsanzeige angegebener Hauptspeicherbereiche auf dem Bildschirm bzw., falls vorher ^P eingegeben wird, zum gleichzeitigen Protokollieren auf dem Drucker.

Das Kommando kann in Verbindung mit DS und SAVE bei Aenderungsarbeiten herangezogen werden (z.B. Wiederherstellen zerstoerter Textdateien). Die entsprechenden Dateien koennen mit LOAD, READ oder READGR in den Speicher gebracht werden.
DUMP und DUMPA werden automatisch beendet, wenn die Dateiendemarke 1AH (^Z) gefunden wird.

Falls mehr Informationen angezeigt werden sollen als mit einem Mal auf dem Bildschirm Platz haben, erfolgt bei gefuelltem Bildschirm ein Stopp.

Weiterarbeitsmoeglichkeiten:

<ENTER> - Anzeige naechste Seite
<Leerzeichen> - naechste Zeile wird "hineingerollt"
beliebige andere Taste - Durchrollen der restlichen Informationen
- Abbruch ist mit ESCAPE-Taste moeglich

DUMP - Anzeige des Speicherbereiches im ASCII-Code (ist der Datei mit Tabulatoren formatiert, dann formatierte Ausgabe)

DUMPX - Anzeige des Speicherbereiches im HEX- und ASCII -Code (in numerierten Zeilen zu je 16 Byte)

DUMPH - Anzeige des Speicherbereiches im HEX-Code (in numerierten Zeilen zu je 16 Byte)

DUMPA - Anzeige des Speicherbereiches im ASCII-Code (in numerierten Zeilen zu je 16 Byte)

Moeglichkeiten der Adressangabe: (DUMPX als Beispiel)

(1)A0=**DUMPX** <ENTER> - Ausgabe von 128 Byte des Speichers
- Start bei Endadresse des letzten DUMP-Kommandos

(2)A0=**DUMPX ,6** <ENTER> - Ausgabe von n Byte ab zuletzt verwendeter Adresse (im Beispiel 6 Byte)

(3)A0=**DUMPX ,,** <ENTER> - Ausgabe gesamter Speicher ab zuletzt verwendeter Adresse

(4)A0=**DUMPX 100** <ENTER> - Ausgabe des Bytes, das auf der angegebenen Adresse steht (im Beispiel 100H)

(5)A0=**DUMPX 100,0F** <ENTER>- Ausgabe von n Byte in HEX (im Beispiel 0FH=15 dezimal) ab angegebener Adresse (im Beispiel 100H)

(6)A0=**DUMPX 100,** <ENTER> - Ausgabe von 128 Byte ab angegebener Adresse (im Beispiel 100H)

(7)A0=**DUMPX 100,,** <ENTER> - Ausgabe des gesamten Speicherinhaltes ab angegebener Adresse (im Beispiel 100H)

Fuer die Bildschirmausgabe sind die Bildschirmsteuertasten (3.3.1.) zu beachten.

3.4.9. ERA

A0=**ERA**

Das Kommando dient zum Loeschen von Dateien. Es kann mit dem Nummernmenue gearbeitet werden. Nach Auswahl der zu loeschenden Dateien erscheint die Abfrage "**ERASE (Y/N)?**". Waehrend des Loeschens wird der Name jeder ausgewaehlten Datei angezeigt. Versehentliches Loeschen von Dateien kann mit dem RECLAIM-Kommando rueckgaengig gemacht werden. Allerdings ist dies nur bis zum naechsten Schreibzugriff auf die Diskette sinnvoll, weil dann die freigegebenen Bereiche bereits wieder verfaelscht sein koennen. Mit ERA kann der [R]-Parameter angegeben werden, wodurch bei jeder zu loeschenden Datei eine nochmalige Entscheidung ueber die Ausfuehrung moeglich ist. Da dadurch die DIENST-Parametereinstellung R [Forderung einer (Y/N) Antwort vor Kommandbausfuehrung --> siehe LOG-Kommando] auf (ON) gebracht wird, sollte nach Beendigen des Loeschens noch einmal ERA [R] zum Rueckstellen auf (OFF) eingegeben werden.

Beispiel:

```
A0=ERA [R] <ENTER>
A0: 1= DOK      .DOK |  2= ASM      .COM |  3=EDIT      .COM
A0: 4= LINK     .DOK
WAEHLE? 1 3

ERASE (Y/N)? <Y> yes

A: DOK     .DOK   (Y/N) <Y> yes
A: EDIT    .COM   (Y/N) <NO> no
A0=
```

Das Nummernmenue kann beim Loeschen nur einer Datei durch direkte Angabe des Dateinamens nach ERA uebergangen werden. Bei Angabe eines mehrdeutigen Dateinamens werden im Nummernmenue nur diesem Dateinamen entsprechende Dateien angezeigt.

3.4.10. EXIT

A0=**EXIT**

Mit diesem Kommando wird in den SCPX-Grundzustand zurueckgekehrt, wobei die USER-Einstellung erhalten bleibt und das vor dem Aufruf von DIENST aktuelle Laufwerk eingestellt wird. Genau wie bei ^C, mit dem in den DIENST-Grundzustand zurueckgekehrt werden kann, erfolgt ein automatischer Speichertest. Die Anzeige eines "?"-Zeichens bedeutet Fehler, worauf ein umfassender Speichertest eingeleitet werden sollte.

3.4.11. FILL

A0=**FILL (aadr) (eadr) (byte)**

(aadr) - Anfangsadresse Speicherbereich in HEX
(eadr) - Endadresse Speicherbereich in HEX
(byte) - Angabe eines Bytes in HEX

Mit diesem Kommando kann ein Speicherbereich mit einem vorgegebenen Zeichen aufgefuellt werden.

Beispiel: A0=FILL 4000 5000 20 <ENTER>

Es wird genutzt, um vor Ueberarbeitung eines Programmes den Speicher mit Hexanullen aufzufuellen. Es ist guenstig, vor Eingaben oder Einlesen den Speicher mit Space zu fuellen, um bei anschliessender Nutzung von DUMPX klar die Grenzen der eingegebenen Daten zu erkennen. Weiterhin hilft es bei der Manipulierung zerstoerter Diskettenihalte, indem der Speicher mit E5 belegt wird und mit WRITE auf der Diskette die funktionstuechtigen Teile isoliert werden.

3.4.12. GO

A0=GO (Dateibezeichnung) (adr)

(adr) - Speicheranfangsadresse in HEX

GO ist eine spezifische Form des RUN-Kommandos, bei der die gewuenschte Programmdatei auf die angegebene Speicheradresse geladen und anschliessend das Programm gestartet wird.

Beispiel: A0=GO BEI.COM 2000 <ENTER>

Dadurch ist es moeglich, spezielle Programme abzuarbeiten, die nicht die uebliche Anfangsadresse 100H haben. Wenn diese Programme mit einem RET abschliessen, wird zu DIENST zurueckgekehrt. Ausserdem koennen mit GO ueber Tastatur eingegebene Parameter an das abzuarbeitende Programm uebergeben werden. Im Programm muessen sie vom DIENST-Eingabepuffer geholt werden. Die Parameteradresse wird im Register HL bereitgestellt.

Beispiel: A0=GO BEI.COM 2000 B:ZUSA.COM DATEI.NEW

3.4.13. GROUP

A0=GROUP

Das Kommando listet die durchnumerierten 2K-Bloecke auf einer Diskette auf, die jeweils eine Datei bilden. Damit kann herausgefunden werden, wie die Dateien physisch auf der Diskette verteilt sind. Bei der Anzeige erscheint nach dem Dateinamen als erstes die Extent-Angabe (ein Extent = 16 K). Es folgen die Blocknummern. Das GROUP-Kommando kann mit dem Nummernmenue arbeiten.

Beispiel:

```
A0=GROUP B: <ENTER>
B0: 1= MAT     .DOK |  2= TEST     .COM |  3= TKO      .DOC
B0: 4= DIENST .COM |  5= DIENST   .DOK
WAEHLE? 1-3

B:MAT      .DOK  00    01 02
B:TEST     .COM  00    03 04 05 06
B:TKO      .DOK  01    09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16
                       17 18
                 02    19 1A 1B 1C 1D
A0=
```

Das Nummernmenue kann bei Auflisten nur einer Datei durch direkte Angabe des Dateinamens nach GROUP uebergangen werden. Bei Angabe eines mehrdeutigen Dateinamens werden im Nummernmenue nur diesem Dateinamen entsprechende Dateien angezeigt.

Falls eine grosse Datei lange Zugriffszeiten hat, kann mit GROUP eine eventuell vorhandene unguenstige Verteilung auf der Diskette nachgeprueft und durch Kopieren auf eine andere Diskette beseitigt werden. Zum Lesen einzelner 2K-Bloecke einer Datei kann das READGR-Kommando genutzt werden.

3.4.14. JP/EX

A0=JP (adr) (argument)
A0=EX (adr) (argument)

(adr) - Sprungadresse
(argument) - moegliche Uebergabeparameter

Mit diesen beiden Kommandos koennen im Speicher befindliche Programme gestartet werden, indem die Programmanfangsadresse angegeben wird. Ab der angegegebenen Sprungadresse muss ein abarbeitungsfaehiges Programm stehen, das mit RET (oder bei gewuenschtem Warmstart mit BDOS-Funktionsaufruf 0 oder JP 0) endet. Beim Kommando JP wird nach Programmausfuehrung zum SCP-Warmstart (zu Adresse 0) gesprungen. Beim Kommando EX wird nach Programmausfuehrung in den DIENST-Grundzustand zurueckgekehrt.

Beispiel: A0=JP 2000

Bei beiden Kommandos koennen ueber Tastatur eingegebene Parameter an das abzuarbeitende Programm uebergeben werden. Im Programm muessen sie vom DIENST-Eingabepuffer geholt werden. Die Pufferadresse wird im Register HL bereitgestellt.

Beispiel: A0=JP 2000 MAT.DAT

3.4.15.1. LOAD

A0=LOAD (dateibezeichnung) (adr)

(dateibezeichnung) - eindeutiger Bezeichner der zu ladenden Datei
(adr) - Speicheranfangsadresse in HEX

Mit dem LOAD-Kommando kann eine Datei von der Diskette in den Speicher geladen werden. In Verbindung mit dem DS-Kommando (Anzeige/Aenderung vom Speicherinhalt) und dem SAVE-Kommando (Schreiben Speicherinhalt auf Diskette) wird es zum schnellen Aendern von Dateien genutzt.
Vor Ausfuehrung von LOAD muss analysiert werden (mit Hilfe des SIZE-Kommandos), wie gross die zu ladende Datei und der zur Verfuegung stehende TPA-Anwenderbereich (mit Hilfe des LOG-Kommandos) sind, um ein versehentliches Ueberschreiben des Betriebssystems bzw. DIENST-Programmes zu verhindern.

Zu grosse Dateien muessen in Sektionen zerlegt und diese einzeln zur Bearbeitung geladen werden (mit Hilfe von GROUP und READGR).

Nach Ausfuehrung von LOAD wird automatisch die beim Laden erreichte Speicherendadresse und die durch die Datei belegte Anzahl von Sektoren (als noetige Information fuer SAVE vermerken!) angezeigt. Werden .COM-Dateien geladen, die normalerweise ab 100H beginnen, so sollte zur rechnerischen Vereinfachung eine Anfangsadresse K+100H (z.B. 5100H) gewaehlt werden. LOAD bietet die Moeglichkeit, Programme in den Speicher zu laden, wenn deren normaler Speicherbelegungsbereich andersweitig benoetigt wird.

Beispiel: A0=LOAD TEST.COM 5100 (ENTER)

3.4.15.2. SAVE

A0=**SAVE (dateibezeichnung) (adr) (sekt)**

(dateibezeichnung) - eindeutiger Bezeichner der aufzuzeichnenden Datei
(adr) - Speicheranfangsadresse in HEX
(sekt) - Anzahl der zu rettenden 128 Byte-Sektoren (Achtung: Unterschied zum residenten Kommmando SAVE!)

Mit SAVE kann der Speicherinhalt ab der angegebenen Adresse auf Diskette aufgezeichnet werden.
Wurde eine Datei von der gleichen Diskette geladen und bei Aenderungen nicht vergroessert, so braucht bei Beibehalten des urspruenglichen Dateibezeichners keine Sektoranzahl angegeben werden. Die urspruengliche Datei wird ueberschrieben. Somit koennen in Verbindung mit LOAD und DS schnell Dateiaenderungen ausgefuehrt werden.

In allen anderen Faellen, wie z.B. bei Nutzung eines anderen Dateibezeichners als der der geladenen Datei, muss die Anzahl der auszugebenden 128-Byte-Sektoren angegeben werden. Dabei kann auf die bei LOAD erfolgte Meldung ueber die eingelesene Sektoranzahl zurueckgegriffen werden.

Beispiel: A0=SAVE TEST.COM 5100 40 (ENTER)

Es ist zu beachten, dass bei DIENST-Kommandos, die das Nummernmenue nutzen, der Speicherinhalt verfaelscht werden kann. So sollten zwischen einem LOAD-SAVE-Vorgang diese Kommandos nicht genutzt werden. Anwendbar sind aber die DIENST-Kommandos DUMP, DS, FILL, MOVE und SEARCH.

3.4.16. LOG

A0=**LOG**

Mit diesem Kommando koennen die aktuellen DIENST-Standardparametereinstellungen angezeigt werden.

Beispiel:

```
A0=LOG
C          If file exists: A-overlay, B-back up, C-ask, D-skip
3          colums
P (ON)     paging
R (OFF)    request Y/N on current file     (Q-request new Name)
V (ON)     read after write
S (ON)     show system files
T (OFF)    stop if disk full
M (OFF)    mark copied files
X          list drives A-P if on-line
U          list users 0-15
¤ (ON)     submit ¤¤¤.SUB - A:DIENST
+/-        1-8 or (R)ead/write, (S)ystem/dir, e(X)tra
DIENST     0100h  - (addr)
TPA        (addr) - (addr)  (number) sectors
A0=
```

Das Beispiel zeigt die uebliche Ausgangseinstellung.

Die Aenderung der Parametereinstellung kann entweder direkt in der DIENST-Grundstellung oder als Abschluss einer DIENST-Kommandoeingabe erfolgen.

Die anzugebenden Parameter muessen in eckigen Klammern [] eingeschlossen werden (abschliessende Klammer kann weggelassen werden). Die Angabe mehrerer Parameter gleichzeitig ist moeglich.

Beispiel: A0=[4¤] (ENTER)
Die Spaltenanzahl bei folgenden Bildschirmanzeigen (z.B. Nummernmenue) ist 4. Der Ruecksprung nach Programmausfuehrung in die DIENST-Grundstellung wird ein- oder ausgestellt (Zeichen "¤").

Einige der Parameter haben Schaltercharakter, die ihre Wertung bei Angabe des Parameters jeweils umschalten. Um sie wieder in den Grundzustand zu bringen, muessen sie ein zweites Mal angegeben werden.

Andere Parameter sind nur fuer das augenblickliche Kommando wirksam und schalten ihre Wirkung nach Ausfuehrung des Kommandos selbst ab.

Bedeutung der einzelnen Parameter:

Beispiel:

A,B,C,D,V,T,M,Q- sind COPY-Parameter (siehe 3.4.4.)

3 - Spaltenanzahl (colums) bei Anzeige (moeglich: 1-9)

P - Bildschirmanzeige (ON) - seitenweise
(OFF) - fortlaufend rollend

R (ON) - Forderung einer (Y/N) - Antwort vor Ausfuehrung jeder Operation (siehe 3.4.4.)

S (ON) - Anzeige von Systemdateien wird nicht unterdrueckt

X,U - sind DIR-Parameter (siehe 3.4.5.)

¤ (ON) - nach Programmausfuehrung erfolgt Ruecksprung in die DIENST-Grundstellung (A0=)

+/- - sind SET-Parameter (siehe 3.4.27.)

DIENST 0100H-(addr)
- zeigt den Speicherbereich an, der durch DIENST.COM belegt ist

TPA (addr)-(addr) (number) sectors
- zeigt den Speicherbereich an, der fuer das Erstellen und Bearbeiten von Dateien verfuegbar ist.

Die gewaehlte Standardparametereinstellung kann geaendert werden und mit SAVE als neue DIENST-Version abgespeichert werden (dadurch neue Ausgangseinstellung).

3.4.17. MOVE

A0=MOVE (aadr1) (eadr1) (aadr2)

(aadr1) - Anfangsadresse des Quellbereiches in HEX
(eadr1) - Endadresse des Quellbereiches in HEX
(aadr2) - Anfangsadresse des Zielbereiches in HEX

Mit dem Kommando MOVE koennen zusammenhaengende Speicherbereiche beliebig verschoben werden. Das Kommando transportiert den Inhalt des Speicherbereiches mit der Anfangsadresse (aadr1) und Endadresse (eadr1) byteweise nach dem ab Adresse (aadr2) beginnenden Bereich.

Beispiel: A0=MOVE 1000 1FFF 4000 <ENTER>

Die Endadresse des Zielbereiches wird nicht angegeben, da DIENST aus der Anfangs- und Endadresse des Quellbereiches die Laenge des zu verschiebenden Bereiches ermittelt. Der Inhalt des Quellbereiches bleibt erhalten, vorausgesetzt, dass er sich nicht mit dem Zielbereich ueberlappt (wenn die Zieladresse niedriger ist als die Endeadresse des Quellbereiches).

Es ist zu beachten, dass MOVE keine Speichergrenzen erkennt, so dass ein Ueberschreiten abgesichert werden muss.

3.4.18. READ

1. Lesen in den Speicher

A0=**READ (track) (sector) (adr) (n sectors)**

(track) - Spur auf Diskette, ab der gelesen wird in DEZ
(sector) - Sektor dieser Spur in DEZ
(adr) - Adresse, ab der das Gelesene im Speicher abgelegt werden soll
(n sectors) - Anzahl der zu lesenden Sektoren

Mit diesem Kommando koennen Daten einer Diskette (ab Spur/Sektor) ab angegebener Adresse in den Speicher eingelesen werden. In Verbindung mit den Kommandos DS und WRITE kann schnell eine Datei geaendert werden. Waehrend dieses Vorganges sollten keine Kommandos genutzt werden, die das Nummernmenue aufrufen, da dadurch eingelesene Daten ueberschrieben werden koennen. Fehlt die Angabe der Sektoranzahl (n sectors), so wird nur ein Sektor eingelesen. Wird die Speicheradresse (adr) nicht angegeben, wird der Sektor ab Adresse 80H eingelesen.

Beispiel:

```
A0=READ 10 4          <ENTER>
A0=READ 10 4 1000     <ENTER>
A0=READ 10 4 1000 8 <ENTER>
```

Waehrend Ausfuehrung des Kommandos erscheinen auf dem Bildschirm eine Reihe von Informationen:
G=XX:YY T=XXX S=XXX PS=XXX At=XXXX-XXXX

G - XX - Blocknummer
YY - Ordnungsnummer im Block
T - Spurnummer
S - Sektornummer
PS - physische Sektornummer
At - Speicherbereich, in den DIENST die eingelesenen Sektoren eingelesen hat

2. Lesen auf Bildschirm

A0=READ (track) (sector) (kind of read) (n sectors)

(track) - Spur, ab der gelesen wird in DEZ
(sector) - Sektor dieser Spur
(kind of read) - Anzeigeform:
XX - Anzeige in HEX und ASCII
XA - Anzeige in ASCII
XH - Anzeige in HEX
X - gleiche Wirkung wie XX
(n sectors) - Anzahl der zu lesenden Sektoren
(soll nur ein Sektor gelesen werden, kann die Angabe entfallen)

Mit diesem Kommando koennen Daten einer Diskette gelesen und auf Bildschirm dargestellt werden. Es wird aber immer nur ein Sektor in den Speicherbereich 80H-0FFH abgelegt. Analog wie bei 3.4.18.1. werden auf dem Bildschirm zusaetzliche Informationen (G,T,S,PS,At - siehe 3.4.18.1.) angezeigt.

Beispiel: A0=READ 10 2 XX 3

3.4.19. WRITE

A0=WRITE (track) (sector) (adr) (n sectors)

(track) - Spur, ab der geschrieben wird in DEZ
(sector) - Sektor dieser Spur in DEZ
(adr) - Speicheradresse, ab der die Daten ausgegeben werden
(n sectors) - Anzahl der auszugebenden Sektoren

Mit diesem Kommando koennen im Speicher ab Adresse (adr) stehende Daten auf Diskette (Spur, Sektor) geschrieben werden. In Verbindung mit den Kommandos READ und DS kann schnell eine Datei geaendert werden. Waehrend dieses Vorganges sollten keine Kommandos genutzt werden, die das Nummernmenue aufrufen, da dadurch eingelesene Daten ueberschrieben werden koennen. Fehlt die Angabe (n sectors), wird nur ein Sektor geschrieben. Wird die Speicheradresse (adr) nicht angegeben, wird der Sektor ab Adresse 80H (bis Adresse 0FFH) ausgegeben.

Beispiel:

```
A0=WRITE 10 4          <ENTER>
A0=WRITE 10 4 1000     <ENTER>
A0=WRITE 10 4 1000 8   <ENTER>
```

Es ist zu beachten, dass der read only-Schutz bei WRITE nicht wirkt und so jeglicher Bereich (auch der Systembereich einschliesslich Verzeichnis) ueberschrieben werden kann. Falls auf den Verzeichnisbereich geschrieben wird, muss ein ^C folgen.
Im Kommando ist auch eine Ziellaufwerksangabe moeglich!

Beispiel: A0=WRITE B: 10 4 1000 8 <ENTER>

3.4.20. READGR

1. Lesen in den Speicher

A0=READGR (start group) (adr) (n sectors)

(start group) - Blocknummer des zu lesenden Blockes in HEX
(adr) - Adresse des Speichers, auf den Daten gelesen werden
(n sectors) - Anzahl der zu lesenden Sektoren des Blockes in DEZ

Dieses Kommando dient zum Lesen von Datenbloecken in den Speicher. Vor dem Kommando kann mit dem DISK-Kommando (Ermittlung der Groesse und Anzahl der Bloecke) und dem GROUP-Kommando (Ermittlung der Verteilung der Bloecke einer Datei auf Diskette) der zu lesende Block analysiert werden. Nach dem Einlesen sollten keine Kommandos genutzt werden, die das Nummernmenue aufrufen, da dadurch eingelesene Daten ueberschrieben werden koennen.
Fehlt die Angabe der Sektoranzahl (n sectors), wird nur ein Sektor eingelesen. Ist die Speicheradresse (adr) nicht angegeben, wird der Sektor ab Adresse 80H eingegeben.

Beispiel: A0=READGR 1 2000 4 <ENTER>

Waehrend der Ausfuehrung des Kommandos erscheinen auf dem Bildschirm eine Reihe von Informationen (G,T,S,PS,At - siehe 3.4.18.1.).

2. Lesen auf Bildschirm

A0=READGR (start group) (kind of read) (n sectors)

(start group) - Blocknummer des zu lesenden Blockes in HEX
(kind of read) - Anzeigeform:
XX - Anzeige in HEX und ASCII
XA - Anzeige in ASCII
XH - Anzeige in HEX
X - gleiche Wirkung wie XX
(n sectors) - Anzahl der zu lesenden Sektoren in DEZ

Mit diesem Kommando koennen Datenbloecke von der Diskette auf den Bildschirm gebracht werden.

Sektorweise werden die Daten in den Speicherbereich 80H-0FFH uebernommen.

Beispiel: A0=READGR 2 XX 4

3.4.21. WRITEGR

A0=**WRITEGR (group) (adr) (n sectors)**

(group)	- Blocknummer des auf Diskette auszugebenden Blockes in HEX
(adr)	- Speicheradresse, ab der die Ausgabe erfolgt
(n sectors)	- Anzahl der auszugebenden Sektoren in HEX

Das Kommando dient zum Schreiben von im Speicher stehender Daten als Block auf die Diskette.
Blocknummer 0 ist fuer das Verzeichnis reserviert. Falls auf den Verzeichnisbereich geschrieben wird, muss anschliessend ein ^C erfolgen. Fehlt die Angabe (n sectors), wird nur ein Sektor ausgegeben. Wird die Speicheradresse (adr) nicht angegeben, wird der Sektor ab Adresse 80H ausgegeben (**Achtung: zwischen READGR und WRITEGR keine Diskettenverzeichniskommandos ausfuehren, da sonst der Bereich ab 80H verfaelscht wird!**).

Das Kommando WRITEGR beruecksichtigt keine Diskettenmerkmale oder Verzeichnis-Dateischutzmerkmale. Deshalb grosse Vorsicht!

Beispiel: A0=WRITEGR 1 2000 4 <ENTER>

3.4.22. RECLAIM

A0=**RECLAIM**

Das Kommando dient dazu, geloeschte Dateien wieder zugaenglich zu machen. Nach Aufruf von RECLAIM werden nacheinander die Namen geloeschter Dateien mit einer (Y/N)-Entscheidungsabforderung angezeigt. Bei Eingabe von Y wird die entsprechende Dateiverzeichniseintragung regeneriert.
Wurde der zu einer geloeschten Datei gehoerige Diskettenbereich bereits mit anderen Daten ueberschrieben, erscheint bei Anzeige des Dateinamens eine entsprechende Mitteilung. Die Datei kann nicht regeneriert werden.
Wurde die Datei gefunden und reorganisiert, kann mit der ESCAPE-Taste das weitere Anzeigen der zu loeschenden Datei abgebrochen werden.

Ist die Dateibezeichnung der wiederherzustellenden Datei bekannt, kann die Regenerierung in der Form:

A0=**RECLAIM (dateibezeichnung)** <ENTER>

erfolgen.

Jeder wiederhergestellten Datei wird aus Sicherheitsgruenden R/O-Status zugewiesen.
***.SUB-Dateien sollten niemals regeneriert werden (dies kann zur Zerstoerung des Diskettenihaltes fuehren). Wird dies versehentlich getan, ist ***.SUB in den Status R/W zu setzen und mit ERA zu loeschen.

Soll eine Datei wiederhergestellt werden, deren Dateibezeichnung bereits auf dieser Diskette existiert, muss die existierende Datei umbenannt werden (Zuweisung eines neuen Namens fuer eine geloeschte Datei innerhalb RECLAIM ist nicht moeglich).

Beispiel:

```
A0=RECLAIM <ENTER>
A0:TEST     .DOC     recover (Y/N)? <n> no
A0:DU       .DOC     can not recover, file with bad extent
A0:LINK     .DOK     recover (Y/N)? <y> yes
```

3.4.23. REN

A0=**REN**

Das Kommando dient zum Umbenennen von Dateien. Da mit dem Nummernmenue gearbeitet wird, ist ein effektiveres Arbeiten als mit dem residenten Kommando REN moeglich, wenn mehrere Dateien umbenannt werden sollen.
Nach der Auswahl der umzubenennenden Dateien werden diese einzeln angezeigt, und es erscheint jeweils die Aufforderung zur Eingabe des neuen Namens (new NAME:). Nach dessen Eingabe erscheint auf der Folgezeile noch einmal alter und neuer Name (alter Name = neuer Name). Das Umbenennen erfolgt nur, wenn die Diskette R/W-Status hat. Andernfalls erfolgt eine Fehlermitteilung, dass nicht auf die Diskette geschrieben werden kann und es muss ein ^C ausgefuehrt werden. Soll eine ausgewaehlte Datei nicht umbenannt werden, wird bei der Aufforderung zur Eingabe des neuen Namens (new NAME:) nur <ENTER> eingegeben.
Zur schnellen Eingabe der neuen Dateibezeichner kann fuer gleichbleibende Teile das Sternsymbol (*) verwendet werden.

Beispiel:

```
A0=REN <ENTER>
A0:1= DOK      .DOK | 2= PIP      .DOK | 3= MIN      .DOK
A0:4= LINK     .DOK | 5= DIENST   .MAC
WAEHLE? 1 3 <ENTER>

A0:DOK      .DOK=new NAME:NEU.TXT <ENTER>
A0=DOK      .DOK=NEU      .TXT
A0:MIN      .DOK=new NAME:K*.* <ENTER>
A0:MIN      .DOK=KIN      .DOK
A0=
```

Durch Eingabe des Parameters [R] ist eine zusaetzliche Namensaenderungsentscheidung (Y/N) nach Eingabe des neuen Namens moeglich (**Achtung: Wirkung bleibt auch bei darauffolgenden Kommandos erhalten. Deshalb durch nochmalige Eingabe [R] den Standard zuruecksetzen**).

Beispiel:

```
A0=REN[R] <ENTER>
A0:1= DOK      .DOK | 2= PIP      .COM | 3= MIN      .DOK
WAEHLE? 1 <ENTER>

A:DOK      .DOK=new NAME:NEU.TXT <ENTER>
A:DOK      .DOK=NEU      .TXT rename (Y/N)? <y>
A0=[R] <ENTER>
A0=
```

Soll nur eine Datei umbenannt werden, kann unter Umgehung des Nummernmenues der alte Name sofort eingegeben werden.

Beispiel:

```
A0=REN PIP.COM <ENTER>
A:PIP      .COM=new NAME:PROG.COM <ENTER>
A:PIP      .COM=PROG     .COM
A0=
```

Generell kann auch jedes andere Laufwerk direkt angesprochen werden (z.B. A0=REN B:).

Es besteht die Moeglichkeit, durch Angabe eines Doppelpunktes (:) nach new NAME:: bei der Eingabe des neuen Dateinamens mehrere ausgewaehlte Dateien ohne Angabe jedes einzelnen neuen Dateinamens umzubenennen, indem beim neuen Dateibezeichner der Teil, der unveraendert bleibt, mit * angegeben wird.

Beispiel:

```
A0=REN B: <ENTER>
B0: 1= DOK      .DOK | 2= PIP      .COM | 3= MIN      .DOK
B0: 4= LINK     .DOK | 5= DIENST   .COM
WAEHLE? 1 3 4

B:DOK      .DOK=new NAME::*.TXT <ENTER>
B:DOK      .DOK=DOK      .TXT
B:MIN      .DOK=MIN      .TXT
B:LINK     .DOK=LINK     .TXT
A0=
```

Ausserdem soll auf die Moeglichkeit hingewiesen werden, dass beim COPY-Kommando durch Angabe des [Q]-Parameters ein sofortiges Umbenennen der Zieldatei beim Kopieren moeglich ist, wodurch auch auf der gleichen Diskette ein Duplikat der Datei erstellt werden kann.

3.4.24. RESET

A0=RESET

Dieses Kommando bietet die Moeglichkeit, ein einzelnes Laufwerk ohne Nutzung des Warmstartes (^C) wieder in den Grundzu-

stand zu setzen (on-line). Somit brauchen nicht wie beim Warmstart alle Laufwerke in den Grundzustand gesetzt werden, wodurch Zeit gewonnen wird und andere R/O-gesetzte Laufwerke nicht ihren Schreibschutz verlieren.

Dieses Kommando kann auch genutzt werden, um Disketten mit zerstoertem Verzeichnis zu retten.

Beispiel:

Das aktuelle Laufwerk ist A. Es soll auf das Laufwerk B kopiert werden, auf dem die Diskette gewechselt wurde (nach dem letzten Warmstart). Um die R/O-Fehlermeldung beim Kopierversuch zu vermeiden, muss vorher A0=RESET B: (ENTER) eingegeben werden.

3.4.25. RUN

A0=**RUN**

Das Kommando dient zum Abarbeiten und Laden von .COM-Dateien (abarbeitbare Programmdateien). Es kann mit dem Nummernmenue gearbeitet werden, das auf der Diskette alle vorhandenen Dateien vom Typ .COM auswaehlt und anzeigt. Nach Auswahl einer Programmdatei und deren Abarbeitung wird wieder in den DIENST-Grundzustand zurueckgekehrt.

Beispiel:

```
A0=RUN B: (ENTER)
B0: 1= INIT     .COM | 2= SYSG     .COM | 3= SEPR     .COM
B0: 4= PIP      .COM | 5= STAT     .COM
WAEHLE? 1 (ENTER)
```

Das zu ladende und abzuarbeitende Programm kann sofort angegeben werden. In diesem Fall kann als zusaetzlicher Parameter der Name jener Datei angegeben werden, die anschliessend durch das geladene Programm bearbeitet werden soll.

Beispiel:

Verwendung beim Textprozessor TP

A0=RUN TP B:TEST.DOK

Der Textprozessor wird vom Laufwerk A geladen, gestartet und es kann die Textdatei TEST.DOK auf Laufwerk B bearbeitet werden.

Hinweis: DIENST ist mit einem automatischen Restart ausgestattet (Submit-Funktion *(ON)/(OFF) ->siehe LOG). Ist dieser eingeschaltet (*(ON)), wird vor dem Starten des gewaehlten Programms (RUN) eine Submit-Datei auf der Diskette geschaffen zur Kontrolle ueber SCP.

Am Programmende wird diese Datei durch DIENST geloescht und DIENST neu gestartet.
Somit muss gewaehrleistet sein, dass die Diskette,von der man Programme mit RUN startet, nicht R/O gesetzt sind, wenn der Parameter [*] (ON) ist.
Bei SCPX V 1.2 erscheint nach Ausfuehrung des mit RUN gestarteten Programms die Fehlerlampe und undefinierte Zeichen auf dem Bildschirm. Durch Druecken einer beliebigen Taste kann zum Programm DIENST zurueckgekehrt werden, wobei die Datei DIENST.COM zum erneuten Einlesen im aktuellen Laufwerk zur Verfuegung stehen muss.

3.4.26. SEARCH

A0=SEARCH (aadr) (eadr) (byte) [(byte)]

(aadr) - Anfangsadresse des Speicherbereiches
(eadr) - Endadresse des Speicherbereiches
(byte) - Byte bzw. Bytefolge (Laenge 1 bis 128 Zeichen), die im angegebenen Speicherbereich gesucht werden sollen

Das Kommando dient zum Suchen von Bytefolgen, dargestellt im HEX- oder ASCII-Code. Aufeinanderfolgende Zeichen im HEX-Darstellung sind durch Leerzeichen voneinander zu trennen. Aufeinanderfolgende Zeichen in ASCII-Darstelleung werden lueckenlos hintereinander eingegeben und die Folge am Anfang und am Ende mit "-Zeichen eingeschlossen. Die Eingabe einer gemischten Codefolge (HEX und ASCII) ist moeglich, wobei zwischen HEX- und ASCII-Zeichen ein Leerzeichen stehen muss.

Sind einzelne Zeichen einer Bytefolge nicht bekannt bzw. beliebig, so kann an den entsprechenden Stellen ein Fragezeichen (?) stehen.
Nach Ausfuehrung von SEARCH werden auf dem Bildschirm alle erkannten Codefolgen mit Speicheradresse angezeigt. Wenn keine entsprechende Codefolge im angegebenen Speicherbereich existiert, kehrt DIENST ohne zusaetzliche Meldung in den Grundzustand zurueck (A0=).

Beispiel:

```
A0=SEARCH 1000 1400 "???WORT" C3 ? ? (ENTER)
1004 "SCPWORT" C3 64 70
12F4 "(0D)(0A)(0A) WORT" C3 C4 78
1309 "NEUWORT" C3 64 70
A0=
```

Es ist zu beachten, dass bei Bytefolgen, die Befehle darstellen, HEX-Speicheradressen so einzugeben sind, wie sie im Speicher stehen (erst LOW-, dann HIGH-Teil).

Beispiel:

Es sollen alle Sprungbefehle (unbedingte Absolutspruenge) im Speicherbereich 6100H bis 8000H gefunden werden, die als Zieladresse 6F06H haben:

```
A0=SEARCH 6100 8000 C3 06 6F (ENTER)
```

3.4.27.1. SET [-character]

A0=**SET [-charakter]** (ENTER) character - 1-8,S,R,X

character - Eingabe des Merkmals in den Dateinamen durch Angabe einer Ziffer von 1-8 (Setzen eines Bits an eine der 8 Stellen des Dateinamens)
- Eingabe des Merkmals in den Dateityp durch Eingabe von S - [S]ystem (1.Byte des Dateityps = Byte 9 in Verzeichniseintragung)
(Datei wird Systemdatei)
R - [R]ead/Write (2.Byte = Byte 10 in Verzeichniseintragung)
(Datei wird R/W gesetzt)
X - E[x]tra (3.Byte = Byte 11 in Verzeichniseintragung)
(ohne weitere Bedeutung)

Dieses Kommando dient zum Setzen von Merkmalen an eine bestimmte Stelle der Dateibezeichnung in der Verzeichniseintragung. Dies erlaubt die Gruppierung einer Anzahl von Dateien durch Setzen des gleichen Merkmals in ihren Dateibezeichnern.

Die markierten Dateien werden angezeigt. Auch in der Verzeichniseintragung nicht genutzte Namensstellen koennen markiert werden.

Beispiel:

```
A0=SET [-S] (ENTER)
A0: 1= TEST1    .MAC | 2= DEMO     .DOK | 3= TEST1    .COM
A0: 4= TEST1    .REL | 5= XXX      .COM | 6= TEST1    .DOK
WAEHLE? 1 3 4 (ENTER)

A0: TEST1    .MAC
A0: TEST1    .COM
A0: TEST1    .REL
A0=
```

Anwendung des Setzens von Merkmalen in der Dateibezeichnung

SET kann in Verbindung mit dem Aufruf anderer DIENST-Kommandos verwendet werden. Es wird das Kommando dann jeweils nur mit den Dateien ausgefuehrt, die das geforderte Merkmal besitzen.

Beispiel:

```
A0=COPY [+5] (ENTER)
A0: 1= TEST1    .MAC | 2= TEST1    .COM | 3= TEST1  / .REL
WAEHLE?
```

Es werden im Nummernmenue nur die Dateien angezeigt, bei denen das 5. Merkmal gesetzt ist.

3.4.27.2. SET

A0=**SET** (ENTER)

Mit SET kann geprueft werden, welche Merkmale in den Dateibezeichnungen gesetzt sind. Sollen alle Dateien geprueft werden, ist sofort nach SET *** oder bei WAEHLE? 1- anzugeben.

Beispiel:

```
A0=SET (ENTER)
A0: 1= TEST1    .MAC | 2= DEMO     .DOK | 3= TEST1    .COM
A0: 4= TEST1    .REL | 5= XXX      .COM | 6= TEST1    .DOK
WAEHLE? 1-

A0: TEST1    .MAC    ----I---.---
A0: DEMO     .DOK    --------.---
A0: TEST1    .COM    ----I---.---
A0: TEST1    .REL    ----I---.---
A0: XXX      .COM    --------.---
A0: TEST1    .DOK    --------.---
A0=
```

3.4.27.3. SET [+character] charakter - 1-8,S,R,X

A0=**SET [+charakter]**

Dieses Kommando dient zum Loeschen der mit SET [-character] gesetzten Merkmale. Es werden im Nummernmenue nur die Dateien angezeigt, die das angegebene Merkmal besitzen. Nach Dateiauswahl wird das Merkmal der gewaehlten Dateien zurueckgesetzt.

Beispiel:

```
A0=SET [+5] (ENTER)
A0: 1= TEST1    .MAC | 2= TEST1    .COM | 3= TEST1    .REL
WAEHLE? 1-

A0: TEST1    .MAC
A0: TEST1    .COM
A0: TEST1    .REL
A0=
```

Achtung: SET arbeitet ohne Ruecksicht auf R/O-Status (deshalb vorher ^C ausfuehren!).

3.4.28. SETDIR/SETSYS

A0=**SETSYS**
A0=**SETDIR**

SETSYS wandelt Dateien zu Systemdateien, die bei "Aus" (OFF) gestellten [S]-Schalter (siehe LOG-Kommando) nicht beim DIR-Kommando bzw. im Nummernmenue mit aufgelistet werden. Die SYS-Dateien koennen mit dem SORT-Kommando (als zuletzt angezeigte Dateien) sichtbar gemacht werden (oder [S]-Schalter auf (ON) setzen --) dann sind diese Dateien durch runde Klammern um den Dateibezeichner speziell markiert).

SETDIR setzt das Systemmerkmal wieder zurueck.

Beide Kommandos koennen mit dem Nummernmenue (nur wandelbare Dateien werden angezeigt) oder mit speziellen Dateibezeichnungen arbeiten.

Achtung: Es wird keine Ruecksicht auf R/O-Status genommen (deshalb vorher ^C ausfuehren!).

3.4.29. SETRO/SETWR

A0=**SETRO**
A0=**SETWR**

SETRO setzt Dateien zum Schutz R/O, so dass sie nicht beschrieben werden koennen.

SETWR macht Dateien wieder beschreibbar.

Beide Kommandos koennen mit dem Nummernmenue arbeiten. Bei Eingabe von SETRO werden nur die Dateien angezeigt, die R/W gesetzt sind. Bei Eingabe von SETWR werden nur die Dateien angezeigt, die R/O gesetzt sind.

Die Datei ***.SUB darf niemals R/O gesetzt werden, da dies zur Zerstoerung des Diskettenlnhaltes fuehren kann.

Achtung: Es wird keine Ruecksicht auf R/O-Status genommen (deshalb vorher ^C ausfuehren!).

3.4.30. SORT

A0=**SORT n** (n=1-4)

Dieses Kommando sortiert die Verzeichniseintragungen alphabetisch nach 4 verschiedenen Moeglichkeiten:

n=1 - Sortieren nach Dateinamen
n=2 - Sortieren nach Dateinamen mit Systemdateien zuletzt
n=3 - Sortieren nach Dateityp
n=4 - Sortieren nach Dateityp mit Systemdateien zuletzt

Bei darauffolgender Eingabe von DIR oder eines Kommandos, das das Nummernmenue benutzt, werden die Dateien dann entsprechend sortiert angezeigt.

Mit SORT Ø erhaelt man ein unsortiertes Verzeichnis. Auf der Diskette bleiben generell die unsortierten Verzeichniseintragungen erhalten.
Um die benutzte SORT-Einstellung permanent als Standard zu setzen, muss das modifizierte DIENST mit SAVE auf Diskette gerettet werden.

3.4.31. SIZE

A0=**SIZE**

SIZE liefert eine Statistik ueber die Groesse von Diskettendateien. Das Kommando kann mit dem Nummernmenue oder mit speziellen Dateinamen arbeiten.

Nach Auswahl der entsprechenden Dateien werden die Anzahl der Sektoren, die die Datei belegt, die Anzahl der freien Sektoren im letzten dazugehoerigen Block, die Groesse der Datei in KByte und die laufende Summe der Groessen der gewaehlten Dateien in KByte angezeigt.

Beispiel:

```
A0-SIZE <EN
A0: 1= TEST1      .COM | 2= TEST1    .MAC | 3= DEMO     .DOK
A0: 4= TKD        .DOC
WAEHLE? 1 4 <ENTER>

A0: TEST1     .COM    44 sectors  15 empty   6K    6K
A0: TKD       .DOC   953 sectors  12 empty  55K   61K
A0=
```

Das Kommando kann benutzt werden, um vor dem Kommando COPY die Groesse der zu kopierenden Datei zu bestimmen (bei verschiedenen Diskettenformaten dabei auch das DISK-Kommando mit heranziehen).

3.4.32. SPEED

A0=**SPEED m** (m=0-9)

SPEED ist ein Kommando, mit dem man die Geschwindigkeit der Bildschirmanzeige fuer alle Ausgaben des DIENST-Programmes einstellen kann.

```
m=0 - schnellste Geschwindigkeit
  |
  |
m=9 - langsamste Geschwindigkeit
```

Das Kommando ist besonders beim Auflisten grosser Verzeichnisse sowie bei den Kommandos TYPE und DUMP nuetzlich.

Waehrend der Ausgabe (durch diese Kommandos) kann durch Eingabe von 0 bis 9 die Geschwindigkeit geaendert werden (ohne Rueckkehr in Grundzustand [A0=] und Eingabe SPEED).

Durch vorherige Eingabe von ^P kann die Ausgabe auch gleichzeitig auf den Drucker erfolgen (ohne dass Seitenvorschub erzeugt werden kann).

3.4.33. STAT

A0=STAT

STAT liefert statistische Informationen ueber alle Laufwerke, die on-line sind. Es werden der R/O- oder R/W-Status der Diskette, der belegte Speicherbereich in KByte, der freie Speicherbereich in KByte und die Diskettenkapazitaet in KByte angezeigt. Bei Angabe R/O wurde die Diskette gewechselt, und die gelieferten Informationen gelten noch fuer die ursprueengliche Diskette (deshalb vorher ^C eingeben!).

Beispiel:

```
A0=STAT B: (ENTER)
A0: R/W  Used: 292K  Free:   2K  Capacity: 296K
B0: R/O  Used: 170K  Free: 124K  Capacity: 296K
```

Die Differenz aus (Gesamtkapazitaet - belegter + freier Speicherbereich)) ergibt sich daraus, dass 2K fuer das Dateiverzeichnis genutzt wird.

3.4.34. TEST

A0=TEST

Dieses Kommando dient zum Test von Disketten auf defekte Sektoren. Es werden alle Spuren der Diskette gelesen bzw. ueberprueft, ausser den Systemspuren.

Nach Kommandoaufruf erscheint die Ausschrift "**- disk checksum & test**". Anschliessend wird getestet. Ist die Diskette frei von defekten Sektoren, erscheinen durchweg die Zeichen ******... und am Ende wird die Kontrollsumme der gelesenen Bytes in HEX angezeigt (**checksum ____**).

Wird ein defekter Sektor auf der Diskette erkannt, so werden Blocknummer, Spurnummer, Sektornummer und physische Sektorposition dieses Sektors sofort angezeigt. Danach wird die Diskette weiter getestet. Ist der Test beendet, fragt "**Save?(Y/N)**", ob man die Dateien, die defekte Sektoren enthalten, retten will. Bei Antwort mit "Yes" werden die defekten Sektoren in einer speziellen Datei isoliert (DIENST schafft

eine bad-sector-Datei im Verzeichnis, in der die defekten Sektoren untergebracht werden). Anschliessend wird gefragt, ob man die Dateien sehen will, die die defekten Sektoren enthalten:

"**Show bad files (Y/N)**". Bei Eingabe "Yes" werden diese Dateien angezeigt. Die Datei ========.=== ist die bad-sector-Datei. Danach erscheint die Frage "**Repair (Y/N)**" zur Ausbesserung defekter Sektoren. Bei "Yes" werden die Dateien neu eingelesen und wenn die defekten Sektoren noch lesbar waren, werden sie mit ihrem Inhalt neu ueberschrieben. Sind sie nicht mehr lesbar, werden sie mit 0 aufgefuellt. Dies fuehrt zum Verlust von 128 Byte/Sektor, jedoch werden die Dateien wieder in lesbarer Form verfuegbar.

Als letzte Information wird die Kontrollsumme (**total checksum**) fuer die Diskette angezeigt.

Beispiel:

```
A0=TEST (ENTER)  - disk checksum & test
**********************************************************
**********************************************************
- bad sector on G=003:02 T=003 S=034 PS=120
**********************************************************
**********************************************************
**********************************************************
**********************************************************
**********************************************************
**********************************************************
**********************************************************
**********************************************************
*****************************
Save?(Y/N)Yes
Show bad files(Y/N)Yes
TKO      .DOC
========.===
Repair(Y/N)Yes
G=003:02 T=003 S=034 PS=120
Disk has 1 bad block
checksum 54BB
```

Die Testfunktion wird sofort abgebrochen, wenn sich das Verzeichnis auf defekten Sektoren befinden.

TEST_S

A0=TEST S

Dieses Kommando ist identisch mit dem Kommando TEST, jedoch werden zusaetzlich die Systemspuren getestet.

3.4.35. TYPE/TYPEX/TYPEH/TYPEA

A0=**TYPE**

Dieses Kommando dient zur Ausgabe von Dateien auf den Bildschirm. Mit TYPEH koennen auch .COM Dateien in HEX-Code sichtbar gemacht werden. Es kann mit dem Nummernmenue gearbeitet werden. Fuer die Bildschirmausgabe sind die Bildschirmsteuertasten (siehe 3.3.1.) zu beachten.

TYPE - nur fuer Textdateien, die mit TP erfasst wurden anwendbar
- wenn Text formatiert ist, dann erfolgt formatierte Ausgabe

TYPEA - zeigt Textdateien in numerierten Zeilen zu je 16 Zeichen sektorweise an (unformatiert) mit Angabe von Blocknummer, Spurnummer, Sektornummer, physischer Sektornummer, Dateiname und Extentnummer

TYPEH - zeigt Dateien in HEX-Code in numerierten Zeilen zu je 16 Byte sektorweise an (auch .COM-Dateien); mit Angaben wie TYPEA

TYPEX - zeigt Dateien in HEX- und ASCII-Code jeweils in numerierten Zeilen zu je 16 Byte sektorweise an (auch .COM-Dat.); mit Angaben wie TYPEA

Nach der Datelauswahl werden die gewuenschten Dateien ausgegeben. Wird eine Datelserie hintereinander ausgegeben, so ist ^K einzugeben, wenn die Ausgabe der aktuellen Datei abgebrochen und zur naechsten uebergegangen werden soll.

3.4.36. USER

A0=**USER** i i-Benutzerbereichsnummer (0-15)

Mit diesem Kommando wird der Benutzerbereich fuer die Diskettendateien fuer laufende Operationen festgelegt.
Als Parameter i kann eine Zahl von 0-15 eingegeben werden. Der User 0 ist nach dem Kaltstart eingestellt.

Beispiel:

```
A0=USER 1 <ENTER>
A1=
```

3.4.37. XUSER

A0=**XUSER j** j-Zielnutzerbereichsnummer

Dieses Kommando erlaubt die Uebertragung von Dateien von einem Nutzerbereich zu einem anderen (auf derselben Diskette oder auf ein anderes Laufwerk --) unter Nutzung von COPY).

Beispiel:

```
A1=XUSER 2 <ENTER>
A1-2=COPY <ENTER>
A1: 1= TEST1    .MAC | 2= TKO      .DOC | 3= TEST1    .COM
A1: 4= XXX      .DOK | 5= TEST     .REL
WAEHLE? 2 4

destination drive:A
(C)opy or (M)ove:C
TKO      .DOC
XXX      .DOK
```

->(C)opy - Dateien werden auf neuen Nutzerbereich kopiert, bleiben jedoch im alten erhalten
- auf derselben Diskette, aber auch auf eine andere moeglich

->(M)ove - Dateien werden auf neuen Nutzerbereich verschoben, bleiben im alten nicht erhalten
- nur auf derselben Diskette moeglich

Jede kopierte bzw. verschobene Datei wird angezeigt.
Durch Eingabe von USER 0 kommt man wieder in den Nutzerbereich 0 (A0=).

3.4.38. UR

A0=**URn** <ENTER>

Dieses Kommando erlaubt Sprunge zu speziellen Routinen, die der Anwender zu DIENST zusaetzlich wuenscht. Sind diese installiert, so koennen sie aufgerufen werden.

Der Speicherbereich fuer die Eingabe der Anwendersprunge beginnt bei der DIENST-Startadresse +40H (d.h. 140H).

Fuer jedes UR-Kommando werden 8 Bytes bereitgestellt.

UR1 -> 0140 - 0147H
UR2 -> 0148 - 014FH
UR3 -> 0150 - 0157H
UR4 -> 0158 - 015FH

Die Sprungadressen koennen mit DS eingegeben werden. Anschliessend muss die neue DIENST-Version mit SAVE gerettet werden.
Standardmaessig ist UR1 ein RETURN (C9).

3.5. Fehlermeldungen

1. Can not LOG on B:
Disk log ERROR on B:, logging to drive only

1. keine Diskette in Laufwerk B eingelegt
2. Laufwerk B nicht korrekt verriegelt
3. defekte Diskette oder Diskettenverzeichnis in Laufwerk B
4. beim Schreiben - Zieldiskette in Laufwerk B kann physischen Schreibschutz haben

5. beim Lesen - kein ^C betaetigt, nachdem Diskette in Laufwerk B eingelegt wurde
6. beim Kopieren von einer schreibgeschuetzten Diskette mit dem Parameter [M](ON) kann DIENST die Marken (),() nicht setzen

7. beim Starten des Programms auf einer schreibgeschuetzten Diskette mit RUN mit dem Parameter [S]ubmit (ON) (automatischer Restart) kann die Submit-Funktion keine Datei DIENST.*** auf der Diskette anlegen

2. Incorrect, for list of commands enter: ?

1. Eingabe eines falschen DIENST-Kommandos
2. Kommandoeingabefehler
3. Eingabe von RUN vor Angabe eines ausfuehrbaren Dateinamens vergessen

3. Invalid argument

1. notwendige Adresseingabe oder andere Informationen nach einem Kommando vergessen
2. Eingabe eines Buchstabens statt einer geforderten Zahl
3. bei SAVE-Kommando Dateilaengenangabe vergessen

4. Press ^C, disk is *R/O on B:

1. Diskettenwechsel ohne nachfolgende Eingabe von ^C --> Versuch, auf schreibgeschuetzte Diskette bzw. Laufwerk zu schreiben

5. Disk full on B:

1. beim Kopieren einer Dateiserie kann die Zieldiskette restlos gefuellt werden
   - wenn Parameter [T](OFF) versucht COPY, den Rest der Serie noch auf die Zieldiskette in Laufwerk B zu bekommen, so dass man bei zu grossen Dateien die Fehlermeldung erhaelt, jedoch danach eine kleinere Datei, die noch darauf passt, kopiert werden kann
   - es kann eine Serie dieser Fehlermeldungen erscheinen

2. Beziehen des RUN-Kommandos auf eine gefuellte Diskette, wenn Parameter [*]submit (ON) ist, so dass keine Submit-Datei geschaffen werden kann

6. file is *R/O

Versuch der Aenderung des Dateiinhaltes einer R/O-gesetzten Datei (erst mit SETWR R/W setzen)

7. Bad sector on READ Abort (Y/N)

wenn defekter Sektor auf Datei beim Lesen erkannt wird

8. No files ????????.??? on A:

1. Frage nach Verzeichnis auf einer leeren Diskette
2. Frage nach Verzeichnis in einem leeren USER-Feld

9. Bad name

bei Benennen oder Umbenennen von Dateien mit Bezeichnungen, die nicht von SCP akzeptiert werden.

Darueberhinaus existieren bei einzelnen Kommandos noch spezielle Fehlermeldungen, deren Bedeutung aus dem angezeigten Fehlertext hervorgeht.

Beispiele:

Out of disk bounds	z.B. bei Kommando READGR ist eine zu grosse Blocknummer angegeben, die die Diskettengrenze ueberschreitet.
Can not copy to same drive	Es wird versucht, ohne neue Namenszuweisung (Parameter Q fehlt) auf die Quelldiskette in den selben Nutzerbereich zu kopieren.

4. Komprimieren einer SCP-Datei CONF 1520 (SCPX)

4.1. Einleitung

Das Programm CONF steht in engem Zusammenhang mit dem Transformationsprogramm UTR1, das SIOS-Anwenderdateien unter Regie des Betriebssystems SCP in SCP-Dateien transformiert.
Bei dieser Transformation werden die Saetze bzw. Bloecke der SIOS-Datei stets mit einem neuen SCP-Satz begonnen. Dabei koennen im Anschluss an den tatsaechlichen logischen Satz auf der Diskette unbenutzte und mit dem Zeichen 00h aufgefuellte Bereiche entstehen, die bei der Weiterverarbeitung dem vorangehenden oder dem nachfolgenden Satz zugerechnet werden.
Da diese Bereiche nicht benoetigt werden und in bestimmten Faellen bei der weiteren Dateibearbeitung stoeren, wurde das Programm CONF geschaffen. Es komprimiert die Datei und beseitigt so derartige bei der Dateitransformation entstandene Bereiche durch lueckenloses Ketten der Saetze (Vergl. auch die Beschreibung zu UTR1 1520 (SCPX)).

4.2. Aufrufen des Programmes

Anzeige/Ausschrift	Erlaeuterung
CONF	- Programmaufruf durch Tastatureingabe; - Abschliessen der Eingabe mit <ENTER>; - Programm CONF.COM wird damit geladen;
CONF 1520 (SCPX) V x.x	- Meldeausschrift des Programmes - x.x steht fuer die Versionsnummer;

4.3. Eingabe der Parameter der Quelldatei

Anzeige/Ausschrift	Erlaeuterung
SOURCE DRIVE NAME:_	- Eingabeaufforderung fuer das Laufwerk, in dem sich die mit UTR1 erzeugte Quelldatei befindet. - Eingabe ist nicht mit <ENTER> abzuschliessen. - Laufwerksname wird auf Gueltigkeit geprueft;

Anzeige/Ausschrift	Erlaeuterung
	- Wurde nur <ENTER> bedient, wird das Programm mit Warmstart verlassen.
SOURCE FILE NAME:_	- Eingabe des Namens der Quelldatei; - Eingabe ist mit <ENTER> abzuschliessen, wenn weniger als 17 Zeichen eingegeben werden. - Wurde nur <ENTER> bedient, verzweigt das Programm zur Eingabe des Quellaufwerksnamens.

4.4. Eingabe der Parameter der Zieldatei

Anzeige/Ausschrift	Erlaeuterung
DESTINATION DRIVE NAME:_	- Eingabe des Laufwerksnamens, in dem sich die Zieldatei befindet; - Quellaufwerk und Ziellaufwerk duerfen gleich sein, wobei durch haeufige Kopfpositionierung erhoehte Programmlaufzeiten auftreten - Eingabe ist nicht mit <ENTER> abzuschliessen. - Der Laufwerksname wird auf Gueltigkeit geprueft. - Wurde nur <ENTER> bedient, verzweigt das Programm zur erneuten Eingabe des Namens der Quelldatei.
DESTINATION FILE NAME:_	- Eingabe des Namens der zu erzeugenden Datei; - Vorangestellte Laufwerksbezeichnung sollte nicht angegeben werden. Wenn sie eingegeben war, wird sie ignoriert.

Anzeige/Ausschrift	Erlaeuterung
	- Es wird geprueft, ob die Datei schon vorhanden ist. Existiert sie noch nicht, wird sie angelegt.
FILE EXISTS ALREADY DELETE FILE (Y OR N):_	- Datei soll geloescht werden;
	- Anzeige erscheint, wenn eine Datei mit diesem Namen bereits existiert.
	- Eingabe <ENTER> oder "N": Eingabe verhindert das Loeschen. Das Programm fordert einen neuen Dateinamen.
	- Eingabe "Y": Die Datei wird geloescht und neu angelegt.
	- Vor Eingabe des Laufwerks kann die Diskette gewechselt werden.
	- Ziellaufwerk und Quellaufwerk duerfen gleich sein.
RECORDS END IN 0D/0A (Y OR N):_	- Satzendekennung;
	- Durch UTR1 ist es moeglich, die uebertragenen SIOS-Saetze mit einer Satzendekennung 0D/0A abzuschliessen oder auch nur in der logischen Satzlaenge zu uebernehmen.
	- Eingabe "Y": Die Dateikomprimierung wird gestartet.
	- Wurde nur <ENTER> bedient, wird die Eingabeaufforderung fuer den Namen der zu erzeugenden Datei wiederholt.
	- Eingabe "N": Es wird zur naechsten Eingabe aufgefordert.

Bei urspruenglich geblockten Dateien, die bei UTR1 in der Uebertragungsart 2 behandelt wurden, ist an dieser Stelle auch "N" einzugeben.
Wurde "N" eingegeben obwohl eine Endekennung vorhanden ist, bleibt diese unberuecksichtigt und erhalten. Durch die Angabe der logischen Satzlaenge bei der nachfolgenden Eingabe wird die bestehende Satzstruktur durch das Einfuegen einer weiteren Endekennung verfaelscht.

Anzeige/Ausschrift	Erlaeuterung
LOGICAL RECORD LENGTH (USE BLOCK LENGTH FOR BLOCKED FILES):_	- Eingabe der logischen Satzlaenge; - Bei weniger als 5 Stellen ist mit <ENTER> zu beenden.

Handelt es sich bei der Quelldatei um eine urspruenglich geblockte SIOS-Datei und wurde bei UTR1 mit der Uebertragungsart 0 gearbeitet (Blockung blieb erhalten ohne Einfuegen einer atzendekennung), dann ist an dieser Stelle die logische Blocklaenge einzugeben. Die Blockung bleibt dann auch in der komprimierten (Ziel-) Datei erhalten. Die Bloecke werden durch die Satzendekennung (hier in der Funktion als Blockendekennung) getrennt und lueckenlos gekettet.

Handelt es sich bei der Quelldatei um eine urspruenglich geblockte SIOS-Datei und wurde bei UTR1 mit der Uebertragungsart 2 gearbeitet (Blockung blieb erhalten, jeder Satz im Block mit Satzendekennung), dann ist durch die folgende Berechnung die einzugebende Blocklaenge zu ermitteln:

$$BL = n \times SL + 2(n-1)$$

BL = Blocklaenge fuer Eingabe
n = Anzahl der Saetze im Block
SL = logische Satzlaenge

Durch das Komprimieren wird auch in diesem Falle ein lueckenloses Ketten aller Saetze erreicht, wobei die Blockung aufgehoben wird.
Soll aus einer geblockten SIOS-Datei eine gekettete SCP-Datei unter Verwendung von UTR1 und CONF erzeugt werden, dann ist bei UTR1 die Transformationsart 1 als die guenstigste Variante zu waehlen (es wird entblockt) und bei CONF mit der logischen Satzlaenge zu arbeiten.

Anzeige/Ausschrift	Erlaeuterung
CONDENSE IS WORKING	- Das Komprimieren ist gestartet (Arbeitszustand);

Anzeige/Ausschrift	Erlaeuterung
	- Wird die Diskette waehrend der Arbeit voll, wird durch eine Meldung informiert.

4.5. Programmende

Anzeige/Ausschrift	Erlaeuterung
END OF FILE	- Ende der Datei ist erreicht; - Anschliessend erfolgt erneuter Sprung zum Programmanfang (entsprechend Pkt.3); - Die erzeugte komprimierte Datei wird zusaetzlich mit der Dateiendekennung 1A versehen. Dadurch wird auch ein Auslisten mit dem Kommando TYPE moeglich.

4.6. Fehlernachrichten

Anzeige/Ausschrift	Erlaeuterung
INVALID DRIVE NAME	- Unzulaessiger Laufwerksname;
INVALID FILE NAME	- Datei dieses Namens ist auf der Diskette als Quelldatei nicht vorhanden. - Es ist ein anderer Name einzugeben.
DIRECTORY IS FULL	- Das Verzeichnis der Diskette ist voll. Es besteht die Moeglichkeit, die Diskette zu wechseln, bevor der Laufwerksname eingegeben wird.
R/W ERROR	- Schreibfehler beim Anlegen der Datei. Es kann vor Eingabe des Laufwerksnamens die Diskette gewechselt werden.

Anzeige/Ausschrift	Erlaeuterung
ERROR WRITE DISK	- Fehler beim Schreiben der Zieldatei (oder gefuellte Zielkette). Das Programm bricht mit Bearbeitung ab und kehrt zum Anfang zurueck.

Fehler beim Lesen der Quelldatei werden direkt vom Betriebssystem gemeldet. Es werden nach Bedienen von <ENTER> Lesewiederholungen durchgefuehrt. Bleibt das erfolglos, bricht das Programm die Bearbeitung ab und kehrt zum System zurueck.

4.7. Darstellung der Zusammenhaenge des Dateiaufbaus SIOS und SCP bei der Bearbeitung mit UTR1 und CONF

	UTR1		CONF	
SIOS-Datei	Transformationsart	SCP-Datei	Verarbeitungsart	SCP-Datei
ungeblockt	0	neuer SIOS-Satz = neuer SCP-Satz	in logischer Satzlaenge	Saetze werden gekettet und erhalten eine Satzendekennung
	1			
	2	neuer SIOS-Satz = neuer SCP-Satz mit Endekennung OD/OA (EK)	mit Endekennung	Saetze werden gekettet,EK bleibt erhalten
geblockt	0	Blockung bleibt erhalten; neuer SIOS-Block = neuer SCP-Satz	in logischer BL=(n SL)	Bloecke werden gekettet und erhalten eine EK
	1	entblocken; neuer SIOS-Satz = neuer SCP-Satz	in logischer SL	Saetze werden gekettet mit EK
	2	Blockung bleibt;neue SIOS-Block = neuer SCP-Satz; jeder Satz mit EK	in log. SL/BL BL=nSL+(n-1)	Saetze werden gekettet;EK bleibt erhalten

Beachte:

Soll aus einer geblockten SIOS-Datei eine gekettete SCP-Datei durch die Programme UTR1 und CONF erzeugt werden, dann ist bei UTR1 die Transformationsart 1 als die guenstigste Variante zu waehlen, obwohl die durch UTR1 erzeugte Datei unter Umstaenden groesser ist.
Die Weiterverarbeitung durch CONF ist mit den dargestellten Ergebnissen fuer alle UTR1-Transformationsarten moeglich.

5. Drucken einer SCP-Datei PRIF 1520 (SCPX)

5.1. Einleitung

Durch das Programm wird die Moeglichkeit geschaffen, beliebige SCP-Dateien auf Papier unterschiedlicher Breite zu drucken.

5.2. Aufrufen des Programmes

Anzeige/Ausschrift	Erlaeuterung
PRIF	- Programmaufruf durch Tastatureingabe; - Abschliessen der Eingabe mit <ENTER>; - Programm PRIF.COM wird damit geladen;
PRIF 1520 (SCPX) V x.x	- Meldeausschrift des Programmes - x,x steht fuer die Versionsnummer;

5.3. Eingeben der Dateiparameter

Anzeige/Ausschrift	Erlaeuterung
DRIVE NAME:_	- Eingabeaufforderung fuer das Laufwerk, in dem sich die Diskette mit der zu druckenden Datei befindet. - Eingabe ist nicht mit <ENTER> abzuschliessen; - Laufwerksname wird auf Gueltigkeit geprueft. - Wurde nur <ENTER> bedient, wird das Programm mit Warmstart verlassen.
FILE NAME:_	- Eingabe des Dateinamens (eindeutiger SCP- gueltiger Dateibezeichner) - Eingabe ist mit <ENTER> abzuschliessen, wenn weniger als 17 Zeichen eingeben werden.

Anzeige/Ausschrift	Erlaeuterung
	- Wurde nur <ENTER> bedient, verzweigt das Programm zur Laufwerkseingabe. - Der eingegebene Dateiname wird auf der Diskette gesucht. Wird die Datei nicht gefunden, erscheint eine Fehlermeldung und erneute Eingabe des Dateinamens wird gefordert.
RECORDS END IN OD/OA (Y OR N):_	- Satzendekennung; - Eingabe "Y": Der Drucklistenparameter wird fortgesetzt. - Wurde nur <ENTER> bedient, kehrt das Programm zur Eingabe des Dateinamens zurueck. - Eingabe "N": Aufforderung zur naechsten Eingabe;
LOGICAL RECORD LENGTH:_	- Eingabe der logischen Satzlaenge; - Bei weniger als 5 Stellen ist mit <ENTER> zu beenden; - Wurde nur <ENTER> bedient, kehrt das Programm zur Frage nach der Satzendekennung zurueck.

5.4. Eingeben der Parameter der Druckliste

Anzeige/Ausschrift	Erlaeuterung
PRINT RECORDS IN FULL LENGTH (Y OR N):_	- Es besteht die Moeglichkeit, die Saetze der Datei nur anzudrucken (Teilduck). - Wurde nur <ENTER> bedient, kehrt das Programm zur Frage nach der Satzendekennung zurueck.

Anzeige/Ausschrift	Erlaeuterung
	- Eingabe "Y": Eingabe der Druckzeilenlaenge;
	- Eingabe "N": Durch die Anzeige:
RECORD LENGTH TO PRINT:_	wird zur Eingabe der zu drukkenden Satzlaenge aufgefordert.
	- Werden weniger als 5 Ziffernstellen eingegeben, ist Eingabe mit <ENTER> abzuschliessen.
	- Wurde nur <ENTER> bedient, wird die vorherige Frage wiederholt.
LENGTH OF PRINTING LINE (CHARACTERS/LINE):_	- Eingabe der Druckzeilenlaenge;
	- Es ist ein Wert im Bereich 1...128 einzugeben.
	- Bei Eingabe von 0 oder >128 wird diese Frage wiederholt.
	- Wurde nur <ENTER> bedient, kehrt das Programm zur vorherigen Eingabe zurueck.
1 LINE DISTANCE BETWEEN PRINTED RECORDS (Y OR N):_	- Es besteht die Moeglichkeit, nach jedem gedruckten Satz eine Leerzeile einzufuegen.
	- Wurde nur <ENTER> bedient, kehrt das Programm zur vorherigen Eingabe zurueck.
PRINTER READY? (Y OR N):_	- Druckerbereitschaft;
	- Eingabe "N": Wiederholung der Frage;
	- Wurde nur <ENTER> bedient, kehrt das Programm zur vorherigen Eingabe zurueck.

Anzeige/Ausschrift	Erlaeuterung
	- Eingabe "Y": Kontrolle der Druckerbereitschaft; - Ist der Drucker nicht bereit, erfolgt ggf. eine Fehlermeldung in Abhaengigkeit vom Druckertyp und von dessen Einschaltzustand und die Frage nach der Druckerbereitschaft wird erneut gestellt.
PRINT IS WORKING	- Druck ist gestartet (Arbeitszustand);

Auf dem Papier wird zunaechst ein Kopftext folgenden Inhalts gedruckt:

P R I F 1520 (SCPX) V x.x

PRINT SCP FILE **FILE NAME:nnnnnnnnnnnnnnnn**

wobei x.x fuer die Versionsnummer von PRIF steht und nnn...nnn den Dateinamen darstellt.

Anschliessend wird die Datei in der gewuenschten Form vollstaendig gedruckt.

5.5. Dateiende

Anzeige/Ausschrift	Erlaeuterung
END OF FILE	- Ende der Datei ist erreicht; - Dieser Text wird als Abschluss auf die Liste gedruckt. - Anschliessend kehrt das Programm zum Anfang zurueck.

5.6. Fehlermeldungen

Anzeige/Ausschrift	Erlaeuterung
INVALID DRIVE NAME	- Unzulaessiger Laufwerksname;
INVALID FILE NAME	- Datei dieses Namens ist auf der Diskette nicht vorhanden.

Anzeige/Ausschrift	Erlaeuterung
	- Es ist ein anderer Name einzugeben.
PRINTER IS NOT READY	- Keine Druckerbereitschaft;

Fehler beim Lesen der Diskette werden direkt vom Betriebssystem gemeldet. Es werden nach Bedienen von <ENTER> Lesewiederholungen durchgefuehrt. Bleibt das erfolglos, bricht das Programm die Bearbeitung ab und kehrt zum System zurueck.

6. Wandeln einer SIOS-Textdatei fuer die TP-Bearbeitung TRTX 1520 (SCPX)

6.1. Einleitung

Bei einer Vielzahl von Anwendern besteht der Wunsch, bisher mit dem SIOS-Dienstprogramm TEXT 1520 (SIEX) bearbeitete Texte auch mit dem SCP-Programm TP 1520 (SCPX) bearbeiten zu koennen.
Dazu genuegt es wegen der internen Struktur der von TEXT 1520 (SIEX) verwendeten Datei "TEXT" nicht, nur eine einfache Dateitransformation mittels UTR1 vorzunehmen.

Es ist vielmehr folgender Ablauf einzuhalten:

- Mit Hilfe des SIOS-Programmes TEXT ist die Datei "TEXT" einer REORGANISATION (Kdo. 9) zu unterziehen.

- Es sollte moeglichst nur ein Text in der Datei stehen, da spaeter unter SCP alles als ein Text betrachtet wird. Die uebrigen Texte sind in geeigneter Form zu retten und in der Datei anschliessend zu loeschen.

- Die Uebertragung der Datei "TEXT" erfolgt mittels UTR1 und in der Uebertragungsart 0. Die entstandene SCP-Datei ist zunaechst ein Abbild 1:1 der SIOS-Datei. Es sind neben dem Textverzeichnis auch noch die satzinternen Zeiger und Steuerzeichen vorhanden.

- Die so entstandene Datei wird nun mit dem Programm TRTX 1520 (SCPX) in eine Form gebracht, die die Verarbeitung mit TP 1520 (SCPX) erlaubt.

6.2. Aufrufen des Programmes

Anzeige/Ausschrift	Erlaeuterung
TRTX	- Programmaufruf durch Tastatureingabe; - Abschliessen der Eingabe mit <ENTER>; - Programm TRTX.COM wird damit geladen;
TRTX 1520 (SCPX) V x.x	- Meldeausschirft des Programmes - x.x steht fuer die Versionsnummer;

6.3. Eingabe der Parameter der Quelldatei

Anzeige/Ausschrift	Erlaeuterung
SOURCE DRIVE NAME:_	- Eingabeaufforderung fuer das Laufwerk, in dem sich die mit UTR1 erzeugte Quelldatei befindet; - Eingabe ist nicht mit <ENTER> abzuschliessen; - Laufwerksname wird auf Gueltigkeit geprueft; - Würde nur <ENTER> bedient, wird das Programm mit Warmstart verlassen.
SOURCE FILE NAME:_	- Eingabe des Namens der Quelldatei; - Eingabe ist mit <ENTER> abzuschliessen, wenn weniger als 17 Zeichen eingeben werden. - Wurde nur <ENTER> bedient, verzweigt das Programm zur Eingabe des Quellaufwerknamens.

6.4. Eingabe der Parameter der Zieldatei

Anzeige/Ausschrift	Erlaeuterung
DESTINATION DRIVE NAME:_	- Eingabe des Laufwerksnamens, in dem sich die Zieldatei befindet; - Quellaufwerk und Ziellaufwerk duerfen gleich sein, wobei durch haeufige Kopfpositionierung erhoehte Programmlaufzeiten auftreten koennen. - Eingabe ist nicht mit <ENTER> abzuschliessen. - Laufwerksname wird auf Gueltigkeit geprueft.

- Wurde nur <ENTER> bedient, verzweigt das Programm zur erneuten Eingabe des Namens der Quelldatei.

DESTINATION FILE NAME:_

- Eingabe des Namens der zu erzeugenden Datei;
- Vorangestellte Laufwerksbezeichnung sollte nicht angegeben werden. War sie eingegeben, wird sie ignoriert.
- Es wird geprueft, ob die Datei schon vorhanden ist. Existiert sie noch nicht, wird sie angelegt.

FILE EXISTS ALREADY
DELETE FILE (Y OR N):_

- Datei soll geloescht werden;
- Anzeige erscheint, wenn eine Datei mit diesem Namen bereits existiert.
- Eingabe <ENTER> oder "N": Eingabe verhindert das Loeschen. Das Programm fordert einen neuen Dateinamen.
- Eingabe "Y": Die Datei wird geloescht und neu angelegt.
- Vor Eingabe des Laufwerks kann die Diskette gewechselt werden.
- Ziellaufwerk und Quellaufwerk duerfen gleich sein.

TRANSFORM IS WORKING

- Textumformung ist gestartet (Arbeitszustand);
- Wird die Diskette waehrend der Arbeit gefuellt, wird durch eine Meldung informiert.

6.5. Dateiende

Anzeige/Ausschrift	Erlaeuterung
END OF FILE	- Ende der Datei ist erreicht;
DELETE SOURCE FILE NOW ? (Y OR N):_	- Abfrage, ob die alte Quelldatei sofort geloescht werden soll; - Eingabe "N": Die Datei wird nicht geloescht. - Wurde nur <ENTER> bedient, wird die Frage wiederholt. - Eingabe "Y": Die Quelldatei wird geloescht, und das Programm kehrt an den Anfang zurueck.

6.6. Fehlernachrichten

Anzeige/Ausschrift	Erlaeuterung
INVALID DRIVE NAME	- Unzulaessiger Laufwerksname;
INVALID FILE NAME	- Datei dieses Namens ist auf der Diskette als Quelldatei nicht vorhanden. - Es ist ein anderer Name einzugeben.
DIRECTORY IS FULL	- Das Verzeichnis der Zieldiskette ist gefuellt. Es besteht die Moeglichkeit, die Diskette zu wechseln, bevor der Laufwerksname eingegeben wird.
R/W ERROR	- Schreibfehler beim Anlegen der Datei. Es kann vor Eingabe des Laufwerksnamens die Diskette gewechselt werden.

Anzeige/Ausschrift	Erlaeuterung
ERROR WRITE DISK	- Fehler beim Schreiben der Zieldatei (oder gefuellte Zieldiskette). Das Programm bricht die Bearbeitung ab und kehrt zum System zurueck.
R/W ERROR SOURCE DISK	- Schreibfehler beim Loeschen der Quelldatei. Loeschen ist nicht moeglich. Quelldatei ist nicht geloescht (SOURCE FILE WAS NOT DELETED).

Fehler beim Lesen der Quelldatei werden direkt vom Betriebssystem gemeldet. Nach Bedienen von <ENTER> werden Lesewiederholungen durchgefuehrt. Bleibt das erfolglos, bricht das Programm die Bearbeitung ab und kehrt zum System zurueck.

7. Diskettenanalyseprogramm LESD 1520 (SCPX)

Das Programm stellt die physischen Parameter einer unbekannten Diskette fest und prueft, ob sich alle Sektoren lesen lassen. Der Disketteninhalt wird **nicht** zerstoert!

Anzeige/Ausschrift	Erlaeuterung
LESD	- Programmaufruf durch Tastatureingabe; - Abschliessen der Eingabe mit <ENTER>; - Programm LESD.COM wird damit geladen;
LESD 1520 (SCPX) V x.x TEST DISKETTE	- Meldeausschrift des Programmes - x.x steht fuer die Versionsnummer;
PRINT (Y OR N):_	- Eingabe Y: Protokolldruck wird gewuenscht - Eingabe N: kein Protokolldruck; - Eingabe ist nicht mit <ENTER> abzuschliessen; - Wurde nur <ENTER> bedient, wird das Programm beendet.
LOAD SCP SYSTEM DISKETTE INTO DRIVE A PRESS <ENTER>	- SCP-Systemdiskette in Laufwerk A einlegen und mit <ENTER> abschliessen; - Damit wird das Programm ordnungsgemaess mit Warmstart beendet.

Beachte:
Es sollte keine Eingabeaufforderung zum Zweck des Programmabbruchs mit ^C beantwortet werden, da dann unter Umstaenden der Warmstart des Systems nicht ausgefuehrt werden kann!

Anzeige/Ausschrift	Erlaeuterung
DRIVE (A - E):_	- Eingabeaufforderung fuer das Laufwerk, in dem sich die zu pruefende Diskette befindet. - Eingabe ist nicht mit <ENTER> abzuschliessen; - Wurde nur <ENTER> bedient, verzweigt das Programm zur Druckauswahl; - Eine fehlerhafte Eingabe fuehrt zu der Meldung:
INVALID DRIVE NAME	- Eingabewiederholung richtiges Laufwerk;

Anzeige der physischen Diskettenparameter

Fuer eine Diskette, die dem Format IBM 3740, KROS 5108 bzw. KROS 5110/01 entspricht:

IDENTIFICATION: 8" [5 1/4"] SIOS DISKETTE
SINGLE [DOUBLE] SIDE, SOFT SECTOR: nnnn
SINGLE [DOUBLE] DENSITY
DEFECTIVE TRACKS: ii, jj

Fuer Disketten, die nicht diesem Standard entsprechen:

IDENTIFICATION: 8" [5 1/4"] DISKETTE, SINGLE [DOUBLE] SIDE
TRACK 0: SOFT SECTOR: nnnn,
SINGLE [DOUBLE] DENSITY
OTHER TRACKS: SOFT SECTOR: nnnn,
SINGLE [DOUBLE] DENSITY

nnnn ist die Sektorlaenge und kann sein: 128, 256, 512 oder 1024 Byte.
ii und jj sind die Nummern der defekten Spuren.

Anzeige/Ausschrift	Erlaeuterung
UNDEFINED DISK FORMAT !!!	- Es konnte keines dieser Formate festgestellt werden; - Das Programm geht zurueck zur Druckauswahl.

Nachdem die Diskette identifiziert ist, wird sie in der folgenden Art sektorweise sequentiell gelesen.

Die Arbeit des Programms kann waehrend des Lesens zu jeder Zeit unterbrochen werden - ausser beim Lesen der Spur 0 - durch das Betaetigen einer beliebigen Taste.

Anzeige/Ausschrift	Erlaeuterung
TEST CONTINUE (C) OR END (E):_	- Eingabe 'C': bewirkt, dass das Programm an der Stelle fortgesetzt wird, wo es unterbrochen wurde. - Eingabe 'E': beendet den Diskettentest. - Das Programm verzweigt zur Druckauswahl. - Es ist der Test der naechsten Diskette moeglich, bzw. das Beenden des Programmes.

Als Ablaufinformation wird auf dem Bildschirm die Nummer der gerade bearbeiteten Spur angezeigt.
Fuer jeden nicht ordnungsgemaess gelesenen Sektor wird eine Fehlermeldung ausgegeben:

TRACK: nn SECTOR: nn Fehlerart

- nn steht fuer die entsprechende Nummer.

- Die Fehlerart kann sein:

  MARKED SECTOR
  INDEX ERROR
  ID FIELD ERROR
  CRC ERROR

Ausserdem kann als Fehlermeldung:

  DRIVE NOT READY
  REPEAT (Y OR N):_

  auftreten.

- Das Laufwerk war nicht bereit.

- Eingabe 'Y':
  Die Leseoperation wird wiederholt;

- Eingabe 'N':
  Das Programm wird beendet.

8. Transformationsprogramm fuer Text 20 - Dateien in Text 30 - Dateien UTR3 1520 (SCPX)

8.1. Einleitung

Das Programm UTR3 1520 (SCPX) transformiert Textdateien, die mit dem Textsystem Text 20 (unter dem Betriebssystem SIOS) erstellt wurden, in Textdateien die mit dem SCP- Textsystem TP weiterbearbeitet werden koennen.
Damit wurde die Moeglichkeit geschaffen, eine problemlose Weiterbearbeitung der Text 20 - Dateien unter TP (SCPX) vorzunehmen.
Alle transformierten Text 20 - Dateien werden fuer TP im A4-Hochformat ohne Begrenzung des rechten Randes abgelegt, unabhaengig, in welchem Format die Dateien im Text 20 - System erfasst wurden. Es wurde der Grundsatz gewahrt:

eine Zeile im Text 20 = eine Zeile in TP
eine Seite im Text 20 = eine Seite in TP

Transformiert wird pro Durchlauf eine komplette Text 20 - Datei. Da Text 20 mit einem deutschen Zeichensatz arbeitet, TP aber allgemein mit dem Latein-Standard-Zeichensatz, ist eine Auswahlmoeglichkeit im UTR3 vorhanden, eine Umwandlung in ae,oe,ue und ss zu bestimmen.
Die Text 20 - Steuerzeichen Fettdruck, Schattendruck, Unterstreichen, Index, Exponent, Rueckschritt, Trennzeichen und Zeilenendezeichen werden in analoge TP - Steuerzeichen umgewandelt. Geschuetzte Leerzeichen werden als Space (20H) ausgegeben. Alle anderen Zeichen werden 1 : 1 uebernommen.

Eine transformierte Datei wird in den haeufigsten Faellen nochmals ueberarbeitet werden muessen.

Die Zusammenarbeit zwischen UTR3 und dem Bediener ist wie folgt gestaltet:

Anzeige/Ausschrift	Erlaeuterung
UTR3 <ENTER>	- Aufruf von UTR3 durch Eingabe vom Bediener. UTR3.COM ist damit geladen.
UTR3 1520 (SCPX) V x.x	- UTR3 Meldeausschrift x.x ist die Versionsnummer

8.2. Parametereingabe fuer die Text 20 - Diskette

Anzeige/Ausschrift	Erlaeuterung
TRANSFER TEXT 20 --> TEXT TP	- Mit Ausschrift der Transformationsrichtung meldet sich das Programm;

Anzeige/Ausschrift	Erlaeuterung
SOURCE DRIVE (TEXT 20 - DISKETTE):_	- Eingabeaufforderung fuer das Quellaufwerk; Buchstabe B bis E fuer das Laufwerk, in dem die Text 20 - Diskette liegt, eingeben. Eingabe nicht mit <ENTER> abschliessen.

Beachte:

Fuer die Text 20 - Diskette darf niemals das Laufwerk A verwendet werden, da UTR3 mit Warmstart (^C) abgebrochen bzw. beendet wird. Damit wird im Laufwerk A immer eine SCP-Diskette mit Betriebssystem verlangt!

- Der Laufwerksname wird auf Gueltigkeit geprueft. Wurde nur <ENTER> eingegeben, wird das Programm mit Warmstart verlassen.

VOLUME ID (TEXT 20 - DISKETTE):_	- Eingabe des VOL-Name der Text 20 - Diskette; Eingabe ist nur dann mit <ENTER> abzuschliessen, wenn der Name kuerzer als 6 Zeichen ist. Datentraegerkennsatz wird gelesen und der eingegebene Name ueberprueft.
	- Wird nur <ENTER> eingegeben, erscheint wieder die Eingabeaufforderung fuer das Quelllaufwerk.
	- Hat Text 20 - Diskette einen anderen Namen, erfolgt Fehlermeldung und erneute Eingabe des Datentraegernamens wird gefordert. Text 20 - Diskette kann vorher bei Bedarf in diesem Laufwerk gewechselt werden.

8.3. Parametereingabe fuer die Text TP - Diskette

Anzeige/Ausschrift	Erlaeuterung
DESTINATION DRIVE (TEXT TP - DISKETTE):_	- Eingabe des Laufwerksnamens fuer die Text TP - Diskette;

Beachte:

Die TP - Diskette muss unter dem Betriebssystem SCPX initialisiert und es muss mittels SYSG ein Betriebssystem darauf gebracht worden sein, wenn sie im Laufwerk A beschrieben werden soll.

- Danach ist der Name der zu erzeugenden TP - Datei nach den in der "Anleitung fuer den Bediener" beschriebenen Regeln einzugeben.
  Eine vorangestellte Laufwerksbezeichnung wird ignoriert.

DESTINATION FILE NAME (TP(SCPX) - DISKETTE):_

- Eingabe Dateiname fuer die TP - Datei;
  Es wird geprueft, ob die Datei schon vorhanden ist. Existiert sie noch nicht, wird die Meldung "NEW FILE" angezeigt und die Datei angelegt.

- Existiert bereits eine Datei dieses Namens, wird gefragt, ob sie geloescht werden soll.

FILE EXISTS ALREADY
DELETE FILE (Y OR N):_

- Eingabe "Y":
  Datei wird geloescht und neu angelegt.

- Eingabe "N" oder <ENTER>:
  Loeschen wird verhindert;
  Programm fordert Eingabe eines neuen Dateinamens;

Anzeige/Ausschrift	Erlaeuterung
	- Ist das Anlegen der Datei nicht moeglich, wird eine Fehlermeldung angezeigt und zur Laufwerkseingabe zurueckgekehrt. Vor der Eingabe der Laufwerksangabe kann die Diskette gewechselt werden.
DESTINATION FILE NAME (TEXT 20 - DISKETTE):_	- Eingabe des Dateinamen der Text 20 - Datei wie er im Verzeichnis bei Text 20 abgelegt ist (max. 30 Zeichen). Ist der Name kuerzer, ist mit <ENTER> abzuschliessen. Leerzeichen sind mit einzugeben!
	- Es erfolgt eine Suche des Dateinamens im Text 20 - Verzeichnis.
	- Wird der Name nicht gefunden, erfolgt die Anzeige:
FILE NAME TEXT 20 NOT FOUND !	
	und es erfolgt der Abbruch der Transformation. Weiterarbeit ab Grundstellung des Programms;
	- Wurde der Dateiname gefunden, erfolgt die Anzeige:
DEUTSCH ---> LATEIN (Y/N):_	- Eingabe "N": Text wird 1:1 uebernommen;
	- Eingabe "Y": Zeichen ae, ue, oe und ss werden eingesetzt;
TRANSFER IS WORKING	- Dateiuebertragung findet statt;

Anzeige/Ausschrift	Erlaeuterung
DESTINATION DISKETTE IS FULL NEW DISKETTE (Y OR N):_	- TP - Diskette ist gefuellt; Entscheidung, ob die Datei auf einer neuen Diskette fortgesetzt werden soll.
END OF TRANSFER	- Dateiuebertragung erfolgreich beendet; - Programm haelt in der Programmgrundstellung; - Die naechste Datei kann uebertragen werden.

8.4. Fehlernachrichten und Meldungen

8.4.1 Fehlernachrichten

Anzeige/Ausschrift	Erlaeuterung
INVALID DRIVE NAME	- Unzulaessiger Laufwerksname; Programm verlangt erneut die Eingabe des Laufwerksnamens;
INVALID VOLUME ID	- Text 20 - Diskette hat einen anderen Namen. Es wird zur Eingabe des Datentraegernamens verzweigt. Der neue Name ist einzugeben. Vorher kann bei Bedarf eine andere Text 20 - Diskette in dieses Laufwerk eingelegt werden.
INVALID SOURCE FILE NAME	- Es wurde eine Diskette eingelegt, die keine Text 20 - Datei enthaelt. Eine andere Diskette muss verwendet werden.
DRIVE NOT READY REPEAT (Y OR N):	- Laufwerk, in dem die Text 20 - Diskette liegt, ist nicht bereit.

Anzeige/Ausschrift	Erlaeuterung
	- Laufwerk bereitmachen und die Eingabeaufforderung mit "Y" oder <ENTER> beantworten. Dann wird die Zugriffsoperation wiederholt. - Wird mit "N" beantwortet, bricht der Transformationsprozess ab und das Programm steht wieder am Anfang. Es kann das Quellaufwerk eingegeben werden. UTR3 kann nur durch <ENTER> verlassen werden.
READ ERROR SIOS-DISKETTE	- Lesefehler auf der Text 20 - Diskette; Verarbeitung wird abgebrochen, das Programm steht wieder am Anfang.
DIRECTORY IS FULL	- Verzeichnis der TP - Diskette ist gefuellt; Es besteht die Moeglichkeit, die Diskette zu wechseln, bevor der Laufwerksname eingegeben wird.
WRITE ERROR SCP-DISKETTE	- Schreibfehler auf der TP- Diskette; Abbruch! Es wird zur Eingabe des Laufwerksnamens fuer die TP - Diskette zurueckgegangen.
READ ERROR SCP-DISKETTE	- Lesefehler auf der TP-Diskette; Abbruch! Es wird zur Eingabe des Laufwerksnamens fuer die TP - Diskette zurueckgegangen.
MEMORY TOO SMALL FOR TRANSFERING	- Der zur Verfuegung stehende freie Hauptspeicher ist kleiner als die Blocklaenge der Text 20 - Datei. Es ist keine Transformation moeglich.

Anzeige/Ausschrift	Erlaeuterung
INVALID FILE NAME	- Der eingegebene TP-Dateiname ist unzulaessig und entspricht nicht dem geforderten SCP-Format. Der Dateiname ist erneut einzugeben.
FILE NAME TEXT 20 NOT FOUND !	- Dateiname fuer die Text 20 - Datei wurde nicht gefunden. Abbruch der Transformation und Halt in der Grundstellung des Programms.

8.4.2. Meldungen

Anzeige/Ausschrift	Erlaeuterung
NEW FILE	- Eine Datei dieses Namens gibt es noch nicht auf der TP-Diskette. Sie wird angelegt.
LOAD SCP-DISKETTE AND PRESS <ENTER>	- Aufforderung, eine neue TP - Diskette in das Laufwerk einzulegen.
FILE EXISTS ALREADY DELETE FILE (Y OR N):_	- Die Datei ist bereits vorhanden. Entscheidung, wie weitergearbeitet werden soll.
DESTINATION DISKETTE IS FULL NEW DISKETTE (Y OR N):_	- Auf der TP- Diskette ist kein Platz mehr vorhanden. Entscheidung, ob die Transformation auf einer neuen TP - Diskette fortzusetzen ist.

Beachte:

Diskette erst wechseln, wenn die Aufforderung dazu erscheint, da die Datei erst abgeschlossen werden muss.